

# 68

## MICRO JOURNAL

Australia A \$ 4.75  
Singapore S \$ 9.45  
Malaysia M \$ 9.45

New Zealand NZ \$ 6.50  
Hong Kong H \$ 23.50  
Sweden 30:-SEK

**\$2.95** USA

### Motorola VME-MACINTOSH-S 50 & Other 68XXX Systems

6809 68008 68000 68010 68020 68030

#### OS-9

The Magazine for Motorola CPU Devices **FLEX**  
For Over a Decade! **SK-DOS**  
A User Contributor Journal

#### This Issue:

"C" User Notes p.8

Mac-Watch p.37

Basically OS-9 p.14

MC68000/68010 Simulator p.38

Interfacing the MC68881 with the MC6809 p.19

And Lots More!

VOLUME IX ISSUE IV • Devoted to the 68XXX User • April 1987

"Small Computers Doing Big Things"

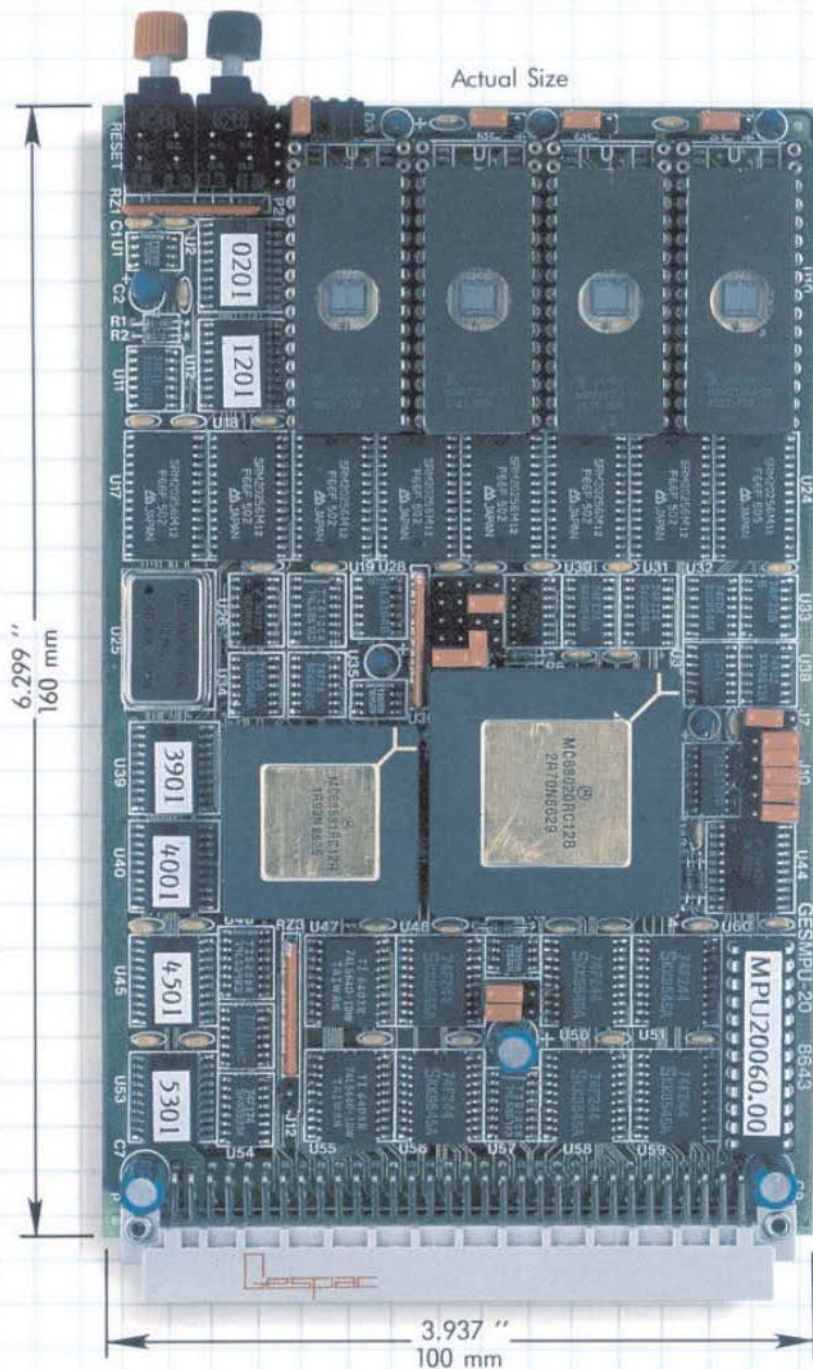
SERVING THE 68XXX USER WORLDWIDE

000+22 A/E  
MR. MICKEY FERGUSON  
P.O. BOX 87  
KINGSTON SPRINGS TN 37082





# GESPAC Gives You More Power Per Square Inch.



Here is the size, performance, and cost breakthrough you have been waiting for: The 68020 based GESMPU-20 from GESPAC.

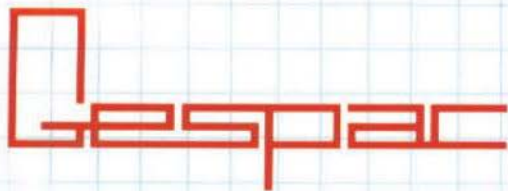
You can now unleash an unprecedented amount of power into your applications. On just 25 square inches, we have squeezed a 12.5 MHz (16 MHz optional) 68020 32-bit microprocessor, a 68881 floating point coprocessor, 4 sockets for up to 512 Kilobytes of EPROM, and up to 512 Kilobytes of zero-wait-states CMOS RAM.

This board is totally expandable through its G-64 bus interface. And GESPAC has the largest variety of inexpensive memory, interfaces, controllers, and transducer cards anywhere. Plus real-time disk operating systems, high level languages, and other software tools. GESPAC has the total solution to your system integration needs.

Best of all, because our boards are small, they cost less. The new GESMPU-20 is priced below \$1000 in one hundred piece quantity orders.

So why wait? Contact us today for information on the GESMPU-20 or any of the 150 G-64 bus system components from GESPAC—the leader in single Eurocard micromputer products worldwide.

**Call (602) 962-5559.**



**IN USA - CANADA**  
50A West Hoover Ave.  
Mesa, Arizona 85202  
Tel. (602) 962-5559  
Telex 386575

**INTERNATIONAL**  
3, chemin des Aulx  
CH-1228 Geneva  
Tel. (022) 713400  
Telex 429989

## GMX MICRO 20 PRICE LIST

MICRO 20 (12.5 MHz) W/1 SAB.....	\$2565.00
MICRO 20 (16.67 MHz) W/1 SAB.....	\$2895.00
MICRO 20 (20 MHz) W/1 SAB.....	\$3295.00

## OPTIONAL PARTS AND ACCESSORIES

68881RC12.....	\$ 295.00
68881RC16.....	\$ 395.00
MOTOROLA 68020 USERS MANUAL.....	\$ 18.00
MOTOROLA 68881 USERS MANUAL.....	\$ 18.00

## SBC ACCESSORY PACKAGE (M20-AP).....\$1690.00

The package includes a PC-style cabinet with a custom backpanel, a Xebec 1410A hard disk controller, a 25 Megabyte (unformatted) hard disk, a 5 1/4" DSDD 80 track floppy disk drive, a 150 watt power supply, cooling fan, panel mounted reset and abort switches and all necessary internal cabling. BACK PANEL PLATE (BPP-PC) For Above.....\$ 44.00  
2nd 5"80 FLOPPY & CABLES FOR M20-AP, ADD.....\$ 250.00  
SECOND 25MB HARD DISK & CABLES, ADD.....\$ 780.00  
TO SUBSTITUTE 65MB HD FOR 25MB HD, ADD.....\$1800.00

## I/O EXPANSION BOARDS

16 PORT SERIAL CARD ONLY (SBC-16S).....\$335.00

The SBC-16S extends the I/O capabilities of the GMX Micro-20 68020 Single-board Computer by adding sixteen asynchronous serial I/O ports. By using two SBC-16S boards, a total of thirty-six serial ports are possible.

RS232 Adapter (SAB-25, SAB-90 or RJ-45).....\$165.00

The board provides level-shifting between TTL level and standard RS-232 signal levels for up to 4 serial I/O ports.

60 LINE PARALLEL I/O CARD (SBC-60P).....\$390.00

The GMX SBC-60P uses three 68230 Parallel Interface/Timers (PI/Ts) to provide up to forty-eight parallel I/O lines. The I/O lines are buffered in six groups of eight lines each, with separate buffer direction control for each group. Buffer direction can be fixed by hardware jumpers, or can be software programmable for bidirectional applications.

PROTOTYPING BOARD (SBC-WM).....\$75.00

The SBC-WM provides a means of developing and testing custom I/O interface designs for the GMX Micro-20 68020 Single-board Computer. The board provides areas for both DIP (Dual Inline Package) and PGA (Pin Grid Array) devices, and a pre-wired memory area for up to 512K bytes of dynamic RAM.

I/O BUS ADAPTER (SBC-BA).....\$195.00

The SBC-BA provides an interface between the GMX Micro-20 68020 Single-board Computer and the Motorola Input/Output Channel (I/O bus). With the I/O bus, up to sixteen off-the-shelf or custom peripheral devices (I/O modules) can be connected to the GMX Micro-20.

ARCNET LAN board w/o Software (SBC-AN)....\$475.00

The SBC-AN provides an interface between the GMX Micro-20 68020 Single-board Computer and the ARCNET modified token-passing Local Area Network (LAN) originally developed by Datapoint Corp. The ARCNET is a baseband network with a data transmission rate of 2.5 Megabits/second. The standard transmission media is a single 93 ohm RG-62/U coaxial cable. Fiber optic versions are available as an option.

OS9 LAN Software Drivers for SBC-AN.....\$ 120.00

## GMX MICRO 20 SOFTWARE

O20 BUG UPDATE - PROMS & MANUAL.....\$ 150.00

THINE 68020 OPERATING SYSTEMS ARE PRICED WHEN PURCHASED WITH THE MICRO-20. PLEASE ADD \$150.00 IF PURCHASED LATER FOR THE UPDATED PROMS AND MANUALS.

## OS9

OS9/68020 PROFESSIONAL PAK.....\$ 800.00  
Includes O.S., "C" EDITOR, ASSEMBLER, DEBUGGER, development utilities, 68881 support.

Other Software for OS-9/68020

BASIC (included in PERSONAL PAK).....\$ 200.00

C COMPILER (included in PROFESSIONAL PAK).....\$ 500.00

PASCAL COMPILER.....\$ 500.00

## UNIFLEX

UniFLEX (when ordered with Board).....\$ 450.00

UniFLEX WITH REAL-TIME ENHANCEMENTS.....\$1000.00

Other Software for UniFLEX

UniFLEX BASIC W/PASCAL COMPILER.....\$ 300.00

UniFLEX C COMPILER.....\$ 350.00

Uni LEX COBOL COMPILER.....\$ 750.00

Uni LEX SCREEN EDITOR.....\$ 150.00

UniFLEX TEXT PROCESSOR.....\$ 200.00

Uni LEX SORT/MERGE PACKAGE.....\$ 200.00

UniFLEX VSAM MODULE.....\$ 100.00

UniFLEX UTILITIES PACKAGE 1.....\$ 200.00

UniFLEX PARTIAL SOURCE LICENSE.....\$1000.00

GMX EXCLUSIVE VERSIONS, CUSTOMIZED FOR THE MICRO-20, OF THE BELOW LANGUAGES AND SOFTWARE ARE ALSO AVAILABLE FROM GMX.

ABSORT FORTRAN (UniFLEX).....\$1500.00

SCULPTOR (specify UniFLEX or OS9).....\$ 995.00

FORTH (OS9).....\$ 595.00

DYNACALC (specify UniFLEX or OS9).....\$ 300.00

GMX DOES NOT GUARANTEE PERFORMANCE OF ANY GMX SYSTEMS, BOARDS OR SOFTWARE WHEN USED WITH OTHER MANUFACTURERS PRODUCT.

ALL PRICES ARE F.O.B. CHICAGO

GMX, Inc. reserves the right to change pricing, terms, and products specifications at any time without further notice.

TO ORDER BY MAIL: SEND CHECK OR MONEY ORDER OR USE YOUR VISA OR MASTER CHARGE. Please allow 3 weeks for personal checks to clear. U.S. orders add \$5 handling if under \$200.00. Foreign orders add \$10 handling if order is under \$200.00. Foreign orders over \$200.00 will be shipped via Emery Air Freight COLLECT, and we will charge no handling. All orders must be prepaid in U.S. funds. Please note that foreign checks have been taking about 8 weeks for collection so we would advise wiring money, or checks drawn on a bank account in the U.S. Our bank is the Continental Illinois National Bank of Chicago, 231 S. LaSalle Street, Chicago, IL 60693, account number 73-32033.

CONTACT GMX FOR MORE INFORMATION ON THE ABOVE PRODUCTS

GMX STILL SELLS GIMIX SSO BUS SYSTEMS, BOARDS & PARTS. CONTACT GMX FOR COMPLETE PRICE LIST.

**A Member of the CPI Family**

# 68 Micro Journal

10 Years of Dedication to Motorola CPU Users

6800 6809 68000 68010 68020

## Editorial Staff

**Publisher**  
Don Williams Sr.

**Executive Editor**  
Larry Williams

**Production Manager**  
Tom Williams

**Office Manager**  
Joyce Williams

**Subscription & Order Desk**  
Lara Prestwood

## Contributing & Associate Editors

Ron Anderson	Dr. E.M. "Bud" Pass
Ron Voights	Art Weller
Doug Lurie	Dr. Theo Elbert
David Lewis	& Hundreds More of Us

## Contents

"C" User Notes	8	Pass
Basically OS-9	14	Voights
MC68881 with the 6809	19	Straub
FORTH	25	Lurie
Ramblings	28	DMW
Mac-Watch	37	DMW
MC68000/68010 SIM	38	Staff
XDMS Review	39	Weller
Bit Bucket	41	All of Us
Classifieds	50	

**68 MICRO JOURNAL**

*"Contribute Nothing - Expect Nothing"* DMW 1986

## COMPUTER PUBLISHING, INC.

*"Over a Decade of Service"*



**68 MICRO JOURNAL**  
Computer Publishing Center  
5900 Cassandra Smith Road  
PO Box 849  
Hixson, TN 37343

Phone (615) 842-4600 Telex 510 600-6630

Copyrighted © 1987 by Computer Publishing, Inc.

68 Micro Journal is published 12 times a year by Computer Publishing Inc. Second Class Postage paid ISSN 0194-5025 at Hixson, TN. and additional entries. Postmaster: send form 3597 to 68 Micro Journal, POB 849, Hixson, TN 37343.

## Subscription Rates

1 Year \$24.50 USA, Canada & Mexico \$34.00 a year.  
Others add \$12.00 a year surface, \$48.00 a year Airmail,  
USA funds. 2 years \$42.50, 3 years \$64.50 plus  
additional postage for each additional year.

## Items or Articles for Publication

Articles submitted for publication must include authors name, address, telephone number, date and a statement that the material is original and the property of the author. Articles submitted should be on diskette, OS-9, FLEX, Macintosh or MS-DOS. All printed items should be dark type and satisfactory for photo-reproduction. No blue ink! No hand written articles - please! Diagrams o.k.

*Please - do not format with spaces any text indents, charts, etc. (source listing o.k.). We will edit in all formatting. Text should fall flush left and use a carriage return only to indicate a paragraph end.*

## Letters & Advertising Copy

Letters to the Editor should be the original copy, signed! Letters of grip as well as praise are acceptable. We reserve the right to reject any letter or advertising material, for any reason we deem advisable. Advertising Rates: Commercial please contact 68 Micro Journal Advertising Department. Classified advertising must be non-commercial. Minimum of \$15.50 for first 15 words. Add \$.60 per word thereafter. No classifieds accepted by telephone.



## The VME BUS and OS-9:

# Ultimate Software for the Ultimate Bus.

Modularity. Flexibility. High Performance. Future growth. These are probably the prime reasons you chose the VME bus. Why not use the same criteria when selecting your system software? That's why you should take a look at Microware's OS-9/68000 Operating System—it's the perfect match for the VME bus.

When you're working with VME you must have access to every part of the system. Unlike other operating systems that literally scream KEEP OUT!, OS-9's open architecture invites you to create, adapt, customize and expand. Thanks to its unique modular design, OS-9 naturally fits virtually any system, from simple ROM-based controllers up to large multiuser systems.

And that's just the beginning of the story. OS-9 gives you a complete UNIX-application compatible environment. It is multitasking, real time, and extremely fast. And if you're still not impressed, consider that a complete OS-9 executive and I/O driver package typically fits in less than 24K of RAM or ROM.

Software tools abound for OS-9, including outstanding Microware C, Basic, Fortran, and Pascal compilers. In addition, cross C compilers and cross assemblers are available for VAX systems under Unix or VMS. You can also plug in other advanced options, such as the GSS-DRIVERS™ Virtual Device Interface for industry-standard graphics support, or the OS-9 Network File Manager for high level, hardware-independent networking.

Designed for the most demanding OEM requirements, OS-9's performance and reliability has been proven in an incredible variety of applications. There's nothing like a track record as proof: to date, over 200 OEMs have shipped more than 100,000 OS-9-based systems.

Ask your VME system supplier about OS-9. Or you can install and evaluate OS-9 on your own custom system with a reasonably priced Microware PortPak™. Contact Microware today. We'll send you complete information about OS-9 and a list of quality manufacturers who offer off-the-shelf VME/OS-9 packages.



### MICROWARE.

#### Microware Systems Corporation

1866 N.W. 114th Street • Des Moines, Iowa 50322  
Phone 515-224-1929 • Telex 910-520-2535

#### Microware Japan, Ltd.

41-19 Honcho 4-Chome, Funabashi City • Chiba 273,  
Japan • Phone 0473 (26) 4493 • Telex 761-299-3122



*Modular Hardware Deserves Modular Software*

Micromaster Scandinavian AB  
S-1 Persgatan 7  
Box 1309  
S-751-43 Uppsala  
Sweden  
Phone: 018-138595  
Telex: 76129

Dr. Rudolf Keil, GmbH  
Porphystrasse 15  
D-6905 Schriesheim  
West Germany  
Phone: (0 62 03) 67 41  
Telex: 465025

Elsoft AG  
Zeilweg 12  
CH-5405 Baden-Dättwil  
Switzerland  
Phone: (056) 83-3377  
Telex: 828275

Vivaway Ltd.  
36-38 John Street  
Luton, Bedfordshire, LU1 2JE  
United Kingdom  
Phone: (0582) 423425  
Telex: 825115

Microprocessor Consultants  
92 Bynya Road  
Palm Beach 2108  
NSW Australia  
Phone: 02-919-4917

Microdata Soft  
97 bis, rue de Colombes  
92400 Courbevoie  
France  
Phone: 1-768-80-80  
Telex: 615405

OS-9 is a trademark of Microware and Motorola. PortPak is a trademark of Microware. GSS-Drivers is a trademark of Graphic Software Systems, Inc. VAX and VMS are trademarks of DEC. Unix is a trademark of AT&T.

# MUSTANG-020 Super SBC™

DATA-COMP proudly presents the first  
Under \$5000 "SUPER MICRO".



## The MUSTANG-020™

### MUSTANG-020™

The MUSTANG-020 68020 SBC provides a powerful, compact, 32 bit computer system featuring the "state of the art" Motorola 68020 "super" micro-processor. It comes standard with 2 megabyte of high-speed SIP dynamic RAM, serial and parallel ports, floppy disk controller, a SASI hard disk interface for intelligent hard disk controllers and a battery backed-up time-of-day clock. Provisions are made for the super powerful Motorola MC68881 floating point math co-processor, for heavy math and number crunching applications. An optional network interface uses one serial (four (4) standard, expandable to 20) as a 125/bit per second network channel. Supports as many as 32 nodes.

The MUSTANG-020 is ideally suited to a wide variety of applications. It provides a cost effective alternative to the other MC68020 systems now available. It is an excellent introductory tool to the world of hi-power, hi-speed new generation "super micros". In practical applications it has numerous applications, ranging from scientific to education. It is already being used by government agencies, labs, universities, business and practically every other critical applications center, worldwide, where true multi-user, multi-tasking needs exist. The MUSTANG-020 is UNIX C level V compatible. Where low cost and power is a must, the MUSTANG-020 is the answer, as many have discovered. Proving that price is not the standard for quality!

As a software development station, a general purpose scientific or small to medium business computer, or a super efficient real-time controller in process control, the MUSTANG-020 is the cost effective choice. With the optional MC68881 floating point math co-processor installed, it has the capability of systems costing many times over it's total acquisition cost.

With the DATA-COMP "total package", consisting of a heavy duty metal cabinet, switching power supply with rf/line by-passing, 5 inch DS/DD 80 track floppy, Xebec hard disk controller, 25 megabyte winchester hard disk, four serial RS-232 ports and a UNIX C level V compatible multi-tasking, multi-user operating system, the price is under \$5000, w/12.5 megahertz system clock (limited time offer). Most all popular high level languages are available at very reasonable cost. The system is expandable to 20 serial ports, at a cost of less than \$65 per port, in multiples of 8 port expansion options.

The system SBC fully populated, quality tested, with 4 serial ports pre-wired and board mounted is available for less than \$3000. Quantity discounts are available for OEM and special applications, in quantity. All that is required to bring to complete "system" stands is a cabinet, power supply, disks and operating system. All these are available as separate items from DATA-COMP.



A special version of the Motorola 020-BUG is installed on each board. 020-BUG is a ROM based debugger package with facilities for downloading and executing user programs from a host system. It includes commands for display and modification of memory, breakpoint capabilities, a powerful assembler/disassembler and numerous system diagnostics. Various 020-BUG system routines, such as I/O handlers are available for user programs.

Normal system speed is 3-4.5 MIPS, with burst up to 10 MIPS, at 16.6 megahertz. Intelligent I/O available for some operating systems.

Hands-on "actual experience sessions", before you buy, are available from DATA-COMP. Call or write for additional information or pricing.

**DATA-COMP**

Installed Systems Worldwide  
OVER 16 YEARS OF DEDICATED QUALITY

**CPI** A Division of  
Computer Products, Inc.  
3000 Commerce Street Road  
Stam, Tx 77743  
Telephone 818 842-4600  
Telex 810 820-0200



## Mustang-020 Mustang-06 Benchmarks

	32 bit Integer	Register Long
IBM AT 7300 Serial Sys 3	9.7	
ATAT 7300 UNIX PC 68010	7.2	4.3
DSC VAX 11/780 UNIX Berkeley 4.2	3.6	3.2
DSC VAX 11/750	5.1	3.2
68000 OS-9 68K 8 MHz	10.0	9.0
68000 OS-9 68K 10 MHz	6.5	6.0
MUSTANG-06 68000 OS-9 68K 10 MHz	9.8	6.3
MUSTANG-020 68020 OS-9 68K 16 MHz	2.7	0.99
MUSTANG-020 68020 MC68001 UNIFLEX 16 MHz	2.6	1.27

```

main()
{
    register long i;
    for (i=0; i < 999999; ++i);
}

```

Estimated MIPS - MUSTANG-020 ..... 4.5 MIPS.  
 Burst to 8 - 10 MIPS: Motorola Speed

### OS-9

OS-9 Professional Ver	\$299.00
*Includes C Compiler	
Basic99	300.00
C Compiler	300.00
Fortran 77	400.00
Microvare Pascal	400.00
OmniGraph Pascal	400.00
Style-Graph	400.00
Style-Spell	195.00
Style-Merge	175.00
Style-Graph-draw-Merge	495.00
PAT w/C source	220.00
JUST w/C source	79.95
PAT/JUST Combo	240.00
Scriptors (see below)	995.00
COM	125.00

### UNIFLEX

UNIFLEX (48620 ver)	\$429.00
Screen Editor	150.00
Sort-Merge	200.00
BASEC/Transpiled	300.00
C Compiler	350.00
COBOL	750.00
CMODEM w/source	100.00
TMODEM w/source	100.00
X-TALK (see Ad)	99.95
Cross Assembler	60.00
Fortran 77	400.00
Scriptors (see below)	995.00

Standard MUSTANG-020™ shipped 12.5 MHz.  
 Add for 16.6 MHz 68020 375.00  
 Add for 16.6 MHz 68881 375.00  
 Add for 20 MHz 68020/68881 750.00

16 Port exp. RS-232 135.00  
 requires 1 or 2 Adapter Cards below.

RS232 Adapter 145.00  
 Each card supports 4 additional ext. ports  
 (total of 24 serial ports supported)

68 line Parallel I/O card 300.00  
 Uses 3 68230 Interface/Timer chips.  
 4 groups of 4 lines each, separate buffer  
 drivers control for each group.

Prototype Board 75.00  
 areas for both dip and PGA devices & a  
 pre-wired memory area up to 512K DRAM.

@PCAN 475.00  
 Interface between the system and  
 ASCHNEY controlled token-passing LAN, fiber optic  
 optional - call.  
 LAN software drivers 120.00

Expander for Motorola I/O Channel Modules 2250.00

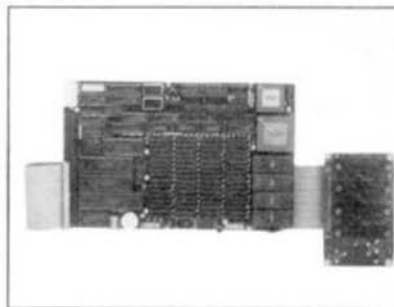
Special for complete MUSTANG-020™ system buyers -  
 \$495.00. SAVE \$300.00

### Software Discounts

All MUSTANG-020™ system and board buyers are entitled  
 to discounts on all listed software: 10-70% depending on  
 item. Call or write for quotes. Discounts apply after the  
 sale as well.

For a limited time we will offer a \$400  
 trade-in on your old 68000 SBC. Must  
 be working properly and complete with  
 all software, cables and documentation.  
 Call for more information.

# THE PRO!



The Mustang-020 board with I/O

**UPGRADES**  
 Write or Call  
 for Professional  
 OS-9 "Full Bore"  
 Upgrade Kit



12.5 MHz (optional 16.6 MHz available) MC68020 full 32-bit wide path  
 32-bit wide data and address buses, non-multiplexed  
 on chip instruction cache  
 object code compatible with all 68000 family processors  
 enhanced instruction set - math co-processor interface  
 68881 math hi-speed floating point coprocessor (optional)  
 direct execution of full 68020 instruction set  
 full support IEEE P754, draft 10.0  
 transcendental and other scientific math functions  
 2 Megabyte of SIP RAM (512 x 32 bit organization)  
 up to 256K bytes of EPROM (64 x 32 bits)  
 4 Asynchronous serial I/O ports standard  
 optional to 20 serial ports  
 standard RS-232 interface  
 optional network interface  
 buffered 8 bit parallel port (U2 MC68250)  
 Centronics type pinout  
 expansion connector for I/O devices  
 16 bit data path  
 256 byte address space  
 2 interrupt inputs  
 clock and control signals  
 Motorola I/O Channel Modules  
 time of day clock/calendar with battery backup  
 controller for 2, 5 1/4" floppy disk drives  
 single or double side, single or double density  
 35 to 80 track selectable (48-96 TPI)  
 SASI interface  
 programmable periodic interrupt generator  
 interrupt rate from micro-seconds to seconds  
 highly accurate time base (3 PPM)  
 3 bit sense switch, readable by the CPU  
 Hardware single-step capability

These hi-speed 68020 systems are presently working at  
 NASA, Atomic Energy Commission, Government  
 Agencies as well as Universities, Business, Labs, and  
 other Critical Applications Centers worldwide, where speed, math  
 crunching and multi-user, multi-tasking UNIX C level V compatibility  
 and low cost is a must!



MUSTANG-020 SBC	\$2490.00
Cabinet w/switching PS	\$299.95
5" 80 track floppy	DS/DD \$269.95
Floppy cable	\$39.95
OS-9 68K Professional Ver.	\$850.00
* Includes C Compiler (\$500.00)	
Whisker cable	\$39.95
Whisker Drive 25 Mbyte	\$895.00
Hard Disk controller	\$395.00
Shipping USA UPS	\$20.00
Total:	\$5,299.80

Save \$1000.00

Complete System Price	\$4,299.80
UNIFLEX	Less \$100.00
MC68881 1/2 math processor	Add \$275.00
16.67 MHz MC68020	\$375.00
16.67 MHz MC68881	\$375.00
20 MHz MC68020 Sys	\$750.00
Note all 68881 chips work with 20 MHz Sys	

complete  
**25 Mbyte HD System**  
**\$4299.80**  
**85 Mbyte HD System**  
**\$5748.80**

Don't be mislead!  
 ONLY Data-Comp  
 delivers the Super  
 MUSTANG-020

**NOTE: Only Professional OS-9  
 now available (68020 Version)  
 Includes (\$500.00) C Compiler  
 68020 & 68881 supported**

# PAT - JUST

**PAT**  
With 'C' Source  
**\$229.00**



**PAT FROM S. E. MEDIA -- A FULL FEATURED SCREEN ORIENTED TEXT EDITOR** with all the best of PIE. For those who swore by and loved PIE, this is for YOU! All PIE features & much more! Too many features to list. And if you don't like ours, change or add your own. C source included. Easily configured to your CRT terminal, with special configuration section. No sweat!

**68008 - 68000 - 68010 - 68020 OS-9 68K \$229.00**

## COMBO PAT//JUST

### **Special \$249.00**

### **JUST**

**JUST from S. E. MEDIA --** Text formatter written by Ron Anderson; for dot matrix printers, provides many unique features. Output formatted to the display. User configurable for adapting to other printers. Comes set-up for Epson MX80 with Graflex. Up to 10 imbedded printer control commands. Compensates for double width printing. Includes normal line width, page numbering, margin, indent, paragraph, space, vertical skip lines, page length, centering, fill, justification, etc. Use with PAT or any other text editor. The **ONLY** stand alone text processor for the 68XXX OS-9 68K, that we have seen. And at a very **LOW PRICE!** Order from: S.E. MEDIA - see catalog this issue.

**68008 - 68000 - 68010 - 68020**  
**With 'C' source**

**OS-9 68K**  
**\$79.95**





An Ace of a System in Spades!

# MUSTANG-08™

## ONE PENNY SALE

NOT 128K, NOT 512K **1¢**  
FULL 768K No Wait RAM



The MUSTANG-08™ system took every hand from all other 68008 systems we tested, running OS-9 68K!

The MUSTANG-08 includes OS9-68K™ and/or Peter Stark's SK\*DOS™. SK\*DOS is a single user, single tasking system that takes up where \*FLEX™ left off. SK\*DOS is actually a 68XXX FLEX type system (Not a TSC product.)

The OS-9 68K system is a full blown multi-user, multi-tasking system. All the popular 68000 OS-9 software runs. It is faster and whiz on disk. The product is: the MUSTANG-08 is faster access than the 68XXX systems are on memory access. Now it is fast! And that is a small part of the! See bench

Intro price of \$1,998.08 (2-80 track floppy) Complete in style cabinet, heavy duty switching power supply, rf by-passing, ready to run, with your choice of OS-9 68K or SK\*DOS. Add \$750 for a single floppy/25 megabyte hard disk system. For those that waited, DATA-COMP didn't forget.

Specifications: System includes OS-9 68K or SK\*DOS - Your Choice

CPU	MC68008	10 Mhz
RAM	768K	256K Chips
	No Wait States	
PORTS	2 - RS232	MC68681 DUART
	2 - 8 bit Parallel	MC6821 PIA
CLOCK	MC146818	Real Time Clock
EPROM	16K, 32K or 64K	Selectable
FLOPPY	WD1772	5 1/4 Drives
HARD DISK	Interface Port	WD1002 Board

Size: 5.75 X 8 inches - bolts directly to a floppy or HD

# Limited Time!

Mustang Hi-Speed Systems  
Only from Data-Comp Div.  
88066-00030

Seconds 32 bit Register  
Integer Long

Other 68008 8 Mhz OS-9 68K...18.0...9.0  
MUSTANG-08 10 Mhz OS-9 68K...9.8...6.3  
Mhz()

C Benchmark Loop

```

/* int i; */
register long i;
for (i=0; i < 9999999; ++i);

```

C Compile times: OS-9 68K. Hard Disk  
file. LIST utility source from K&R.  
MUSTANG-08 0 min - 32 sec  
Other popular 68008 system 1 min - 06 sec  
MUSTANG-020 0 min - 21 sec

Dual 5" Disk System

**\$1,998.08**

25 Megabyte  
Hard Disk System

**\$1,998.09**

Unlike other 68008 systems there are several significant differences. The MUSTANG-08 is a full 10 Megahertz system. The OS-9 NO wait state system. The MUSTANG type performance.

Also, allowing for addressable ROM/PROM the RAM is the maximum allowed for a 68008. The 68008 can only address a total of 1 Megabytes of RAM. The design allows all the RAM space (for all practical purposes) to be utilized. What is not available to the user is required and reserved for the system.

RAM disk 680K can be easily configured, leaving 288K new program RAM space. The RAM DISK can be configured to size your application requires (system must be 28K to 100K to its other requirements). Leaving the user the full program use. Sufficient

FLEX is a trademark of TSC  
OS-9 is a trademark of Microsoft  
MUSTANG-08 is a trademark of CPI  
SK\*DOS is a trademark of Star-K

## Data-Comp Division



A Decade of Quality Service\*

Systems World-Wide

Computer Publishing, Inc. 5800 Cassandra Smith Road  
Telephone 615 842-4601 - Telex 510 600-6630 Murson, Tn 37343

\* Those with SWTPC hi-density FLEX 5" - Call for special info.



*The C Programmers  
Reference Source.  
Always Right On Target!*

## C User Notes

### A Tutorial Series

By: Dr. E. M. 'Bud' Pass  
1454 Latta Lane N.W.  
Conyers, GA 30207  
404 483-1717/4570  
Computer Systems Consultants

#### INTRODUCTION

This chapter concludes the discussion of the proposed ANSI C standard and discusses the enumerated data type, one of the lesser-used features of the C language.

#### PROPOSED ANSI C STANDARD

The proposed ANSI C standard continues the tradition of the K&R version of leaving a large number of details of usage of the language to the implementor. When portability is a concern, as it should be in most cases, these details must be considered. Even when portability is not a concern, considering these implementation-specific items may prevent or solve problems. The following lists of items are taken from the proposed ANSI C standard and are certainly not complete.

The following order dependencies are implementation-dependent:

the order of evaluation of subexpressions, except in certain cases, such as with the && and || operators

the order of side-effects

the order of evaluation of function arguments

the order of memory layout for function arguments and for unrelated declarations

The behavior of the program in the following circumstances is implementation-dependent:

an extern identifier with mutually-incompatible definitions in two source files

the value of a constant not representable in its local context

an undefined escape sequence consisting of a backslash and a lower-case letter

an arithmetic conversion which is not representable in its local context

an arithmetic operation which includes a division by zero or produces a result not representable in its local context

a modification of a const data value by means of a non-const pointer to the const object

a reference to a volatile data value by means of a non-volatile pointer to the volatile object

an array index with value less than zero or greater than the number of items in the array

a reference to an item using a null pointer value

a reference to an item declared locally in a terminated block

an arithmetic operation involving non-array pointers

an arithmetic operation involving pointers not of the same aggregate type

an assignment or data movement involving overlapping operands



the value of a non-initialized operand before assignment

the value of a function for which no value was returned

the value of a function argument in which the declared and actual parameter types are mutually conflicting

the argument to a character handling function out of its domain

the value of a register variable between a setjmp call and its corresponding longjmp call

the number or type of the arguments to a printf or scanf class of function not matching the format string

an invalid conversion character in a printf or scanf format string

the argument to free or realloc not pointing to an area previously allocated by calloc, malloc, or realloc, or previously deallocated by free or realloc

a reference to memory space deallocated by free or realloc

a reference in function registered with the onexit function to data areas allocated by free or realloc

a function registered with the onexit function returns no value

a function registered more than once with the onexit function

The implementor of a conforming C compiler must document the action of the compiler in the following areas:

the values of the arguments to the main function

the number of significant characters in an identifier without external linkage (minimum 31)

the number of significant characters in an identifier with external linkage (minimum 6)

case distinctions and other limitations of external linkage

the number and order of characters comprising an int, short int, long, double, etc.

the number and order of bits in a value of char data type

the characters in the execution character set, their values, and their relationship to the characters in the source character set

the characters in the source character set

an unspecified character in the source text, except in a character constant, a string literal, or in a comment

an unspecified escape sequence, except for a backslash followed by a lower case letter, in a character constant, a string literal, or in a comment

the value of a character constant or string literal containing a character not in the execution character set

the value of a character constant which contains more than one character (if not considered a syntax error)

whether the declaration char is equivalent to signed char or unsigned char

representations and sets of values of all types of ints and floats

the results of converting values of type short int to types signed char and unsigned char

the results of converting values of type int to values of type short int

the results of converting values of type long to values of type int

the results of bitwise operations

the sign of the remainder of a division operation

the result of shifting a value by a negative number of bits or by a value greater than the number of bits in the representation of the value

whether a right shift is considered logical or arithmetic

the direction and type of truncation when a floating point value is converted to a value of type integer

whether the presence of a float data type in the expression causes the entire expression to be evaluated using double data types

the properties of the floating point arithmetic implementation

the result of casting a pointer to an integer and vice versa

the maximum size of an array

the number and types of objects which may be declared with register attribute, and limitations on usage of object with register attribute

the results of accessing union objects of various types

the padding and alignment of members of structures

the order of allocation of fields within a storage unit

whether a field may cross a storage-unit boundary

the maximum number of declarators which may modify a basic type

the maximum number of case values in a switch statement

the method for locating includable source files

the support of quoted names for includable source files

whether a macro name may be the subject of subsequent define preprocessor directives without an intervening undef directive

whether a single-character character constant in a constant expression is considered signed or unsigned

the behavior of each recognized pragma directive

the diagnostic generated by and the termination due to the assert function

the set of characters tested for by the iscntrl and isprint functions

the set of signals for the signal function

whether the default signal handling is reset if the SIGILL signal is received by a function specified by the signal function

the possible values of pid for the kill function

the rules for composing file names and operating system command lines

whether the same file may be open multiple times, either in the same execution or in multiple executions which may be present simultaneously

the effect of the remove function on an open file

the output for the %p conversion in the printf family of functions

the effect of additional mode characters in the fopen function

the details of the value returned from the ftell function

whether a file position in the seek and fseek functions must have been returned from the ftell function

the messages generated by the perror function

the behavior of the abort function (and of program aborts in general) with respect to open and temporary files

the set of environment names and the method of altering the environment list used by the getenv function

whether additional characters other than letters, digits, and underscores, that are not defined in the required source character set, may appear in identifiers

whether string literals are considered as arrays of char, const char, const signed char, etc.

whether a pointer to an object may be cast to a pointer to a function, thus potentially allowing data to be invoked as a function or functions to be inspected or modified

whether types other than int, signed int, and unsigned int may be sub-divided into bit fields

whether the fortran type specifier is allowed for a function to indicate that a function compiled by fortran is to be called

whether the asm statement type is allowed to enable the insertion of assembly-language statements into the source code



whether multiple extern statements referencing the same object in the same source file are allowed, as long as all references are consistent and no more than one reference is initialized

whether macro names, in addition to those in the limits.h source file, are defined to provide the translation and execution environments

### ENUMERATED DATA TYPES

The enum data type is a fairly recent addition to the syntax of the C language. Its use in writing a program is never necessary, so many experienced C programmers have never bothered to learn how to use it. Unfortunately, the implementation of enumerations, even in the proposed ANSI C standard, is not very rich, so there is little incentive for programmers to learn it. Apparently, enumerations were added to C because they exist in Pascal; however, the C language inherently has many features, such as initializers, which are simpler to use and implement features such as enumerations in other manners more compatible with the remainder of the language.

An enumeration is normally a set of sequential values, each of which is termed an enumeration constant. Each of the members of an enumeration may be used as if it were of type const int.

For example, the following enumeration:

```
enum color
{
    red,
    orange,
    yellow,
    green,
    blue,
    indigo,
    violet
}
```

is equivalent to the following enumeration:

```
enum color
{
    red = 0,
    orange = 1,
    yellow = 2,
    green = 3,
    blue = 4,
    indigo = 5,
    violet = 6
}
```

which is equivalent to the following declarations:

```
const int red = 0;
const int orange = 1;
const int yellow = 2;
const int green = 3;
const int blue = 4;
const int indigo = 5;
const int violet = 6;
```

Enumerations are evaluated left to right, so that enumeration initializations such as the following are valid:

```
enum color {red, green = red + 2}
```

Enumeration tags share the same namespace as structures and unions, so they all must be unique within the same source file. Similarly, enumeration constants are in the same namespace as ordinary variables, so they all must be unique within the same source file.

Enumeration tags may be used as type definitions, in which case they are similar to definitions of type int (not const int) and provide no inherent initialization of associated values. Thus, the following declarations are functionally identical:

```
enum color which = blue;
```

and

```
int color = blue;
```

and

```
int color = 4;
```

The sizeof function, when applied to an enumeration tag, returns not the number of elements in the enumeration or the internal length of the ints in the enumeration but the same value as sizeof(int), so it cannot be used to determine the number of elements in the enumeration. Since the only relationship applied to enumeration elements is the default successive initialization values, the equivalents of the Pascal enumeration functions pred and succ are not provided and cannot be defined in general. The enumeration constant and tag names are not directly available for the use of the program; only the values of the enumeration constants are available.

In summary, this is a feature of the C language probably not worth the effort to learn and to use. However, the C programmer should be aware of its existence in case they ever find an application made clearer by its

use or in case they encounter a program written using enumeration constructions.

### C PROBLEM

A BASIC program is being translated (liberated?) into the C language. The following BASIC declaration and statement are encountered:

```
DIM A(10,10)
|
A(10,10) = A(10,10) + 1
```

and are translated into the following C declaration and statement:

```
int a[10,10];
|
a[10,10] = a[10,10] + 1;
```

Describe any errors made in this translation, and provide a corrected translation which maintains both the logic of the original program fragments and the flavor of the C language.

### EXAMPLE C PROGRAM

Following is this month's example C program; it provides routines for manipulating environment variables in UNIX V and UNIX BSD4.x. They will probably require modification to work under OS-9/68000, and may work as-is under 68000 Unixflex, although they have been tested only under UNIX.

```
/*
 * Routines to modify environment
 * variables:
 *
 *   • setenv(name, value) -- sets an
 *   environment
 *   • variable for name to value
 *   • rstenv() -- Resets environment to parent's
 *   • firstenv() -- Initializes sequential scan
 *   • of environment variables
 *   • nextenv() -- Gives next variable
 */

extern char **environ;
/* external variable (provided by system)
 * pointing to parent's environment */
static int initd = 0;
/* set to non-zero after the
 * environment for children is set up */
```

```
static char **p_envron;
/* Holds pointer to original environment
 */
```

```
static int nenv;
/* Number of entries in environment */
```

```
extern char *strcpy(), *getenv();
extern char *malloc(), *realloc();
extern int free();
```

```
/*
 * init() --
 *   • Sets up initd and p_envron
 */
static init()
{
    register int i;

    p_envron = environ;
    for (nenv = 0; p_envron[nenv]; nenv++);
    /* Allocate an array of pointers to
     * each environment string */
    environ = (char **)malloc((unsigned)
        (nenv * sizeof(char **)));
    /* Copy strings to new environment */
    for (i = 0; i < nenv; i++)
    {
        environ[i] = malloc((unsigned)
            (strlen(p_envron[i]) + 1));
        strcpy(environ[i], p_envron[i]);
    }
    /* trailer value */
    environ[nenv] = (char *)NULL;
    initd = nenv++;
}
```

```
#ifdef GETENV
```

```
/*
 * getenv() --
 *   • Returns pointer to value associated
 *   • with name, if any, else NULL;
 *   • a version of this routine is
 *   • present in most standard libraries
 */
```

```
char *getenv(name)
char *name;
{
    register char *v, *s1, *s2;
    register int i;

    for (i = 0; environ[i]; i++)
    {
        for (s1 = environ[i], s2 = name;
            *s1 == *s2; s1++, s2++)
        {
            if (*s2 == '\0' || *s2 == '=')
                /* Reached end of name */
                if (*s1 == '=')
                    return(s1+1);
            else
                break;
        }
        if (*s1 == '=' || *s1 == '\0')
            break;
    }
}
```

```

    }
    }
    return(NULL);
}
#endif /* BSD_4_2 */

/*
 * setenv() --
 *   Sets new value for name
 *   Returns 1 when out of memory,
 *   0 for success
 */
int setenv(name, value)
char *name, *value;
{
    register int i;
    register char *s1, *s2;

    if (initd == 0)
        init();
    for (i = 0; environ[i]; i++)
    {
        for (s1 = environ[i], s2 = name;
            *s1 == *s2; s1++, s2++)
        {
            if (*s2 == '\0' || *s2 == '=')
            {
                /* Reached end of name */
                if (*s1 == '=')
                    goto found;
                else
                    break;
            }
            if (*s1 == '=' || *s1 == '\0')
                break;
        }
    }
    /* not in environment - add it */
    environ = (char **)realloc
        ((char *)environ, (unsigned)
        (++nenv * sizeof(char **)));
    if (environ == (char **)NULL)
        return(1);
    environ[nenv] = (char *)NULL;
    goto out;
found:
    free(environ[i]);
    environ[i] = malloc(strlen(name) +
        strlen(value) + 2);
out:
    strcpy(environ[i], name);
    strcat(environ[i], "=");
    strcat(environ[i], value);
    return(0);
}

/*
 * rstenenv() --
 *   Resets environment to parent's
 */

```

```

rstenenv()
{
    register int i;

    for (initd = 1 = 0; i < nenv; i++)
        free(environ[i]);
    free((char *)environ);
    environ = p_environ;
    nenv = 0;
}

/*
 * firstenv() --
 *   Sets pointer to first variable
 */
static int curenv;

char *firstenv()
{
    if (initd == 0)
        init();
    curenv = 0;
    return(environ[curenv]);
}

/*
 * nextenv() --
 *   Sets pointer to next variable
 */
char *nextenv()
{
    if (initd == 0)
    {
        init();
        curenv = -1;
    }
    return(environ[++curenv]);
}

```

**EOF**

*FOR THOSE WHO NEED TO KNOW*

**68 MICRO  
JOURNAL™**

# Basically OS-9

Dedicated to the serious OS-9 user.  
The fastest growing users group world-wide!  
6809 - 68020

*A Tutorial Series*

By: Ron Voigts  
2024 Baldwin Court  
Glendale Heights, IL  
60139

## Making It Look Nice

This marks the 2nd Anniversary of the column and the beginning of a 3rd year of Basically OS-9. I guess I am a little sentimental. A lot of ground has been covered and there is so much more to go. For every subject I cover, two more come to mind.

I have been reconsidering my goals. There are many people who read this column and the magazine. They range from the hardware hacker to the polished programmer. And everything is covered in between. My intention is to cover something for everyone. I even hope users of the other systems stop in from time to time.

I feel bad that I haven't recently covered anything about BASIC09. I hope the BASIC09 programmers haven't felt left out. I feel that it is an excellent tool for rapid software development and good, sound programming. Many programs in this column are first written in BASIC09 to test their feasibility. Later they are rewritten to be compiled or assembled by some other language. I also develop software with it to solve engineering problems encountered in my classes. And even as an end product, BASIC09 programs are fast and accurate.

One of its strong features is its handling of formatted output. Other languages vary in how well they can handle the formatted output. But BASIC09 can do a very nice job of putting your lines into a form that you want.

I think there are two basic ways a language can handle it. C and many of the Basics require a line that appears like the line to be printed. Variables are represented with markers in locations that they are to appear. The other method is to create a format list of how to print the variables in another list. FORTRAN uses this method. So does BASIC09. The format list contains specifications of how the variables are to be printed. There is a one to one correspondence between the items in the format list and the variable list. Actually, this is a little over simplified it, but we'll get to the variations later. The syntax is:

**PRINT USING <format list>, <variable list>**

The format list is a string. The variables are listed, separated by commas. There are 6 type of format specifications. They are

Rw.fj	Real format
Ew.fj	Exponential format
Iwj	Integer format
Hwj	Hexidecimal format
Swj	String format
Bwj	Boolean format

w	Width of field	1 <= w <= 255
f	Fraction of field	1 <= f <= 9
j	Justification	< (left) > (right) ^ (centered)



The w is for field width. The f specifies how much of the field is to be placed to the right of the decimal point. The justification field is optional and has a few variations. I'll let you consult your manual about then.

A few examples will best illustrate how to use the formatted print.

```
PRINT USING "R5.2>"          3.40
PRINT USING "I8^", 329        00000329
PRINT USING "H4", 62          003E
PRINT USING "S10","HELLO THERE" HELLO THER
```

The first example prints in a field of 5 wide the number 3.4 with 0 padded to the number to make it two places to the right of the decimal. The > means it is right justified. The next one prints 329 in an 8 wide field. The ^ means to right justify it and pad it with leading zeros. (This is one of those variations.) The third prints 62 in hexadecimal. Here leading zeros are automatically added. The last prints a string. Notice the last character is truncated. This is because the field was specified to 10 characters, but the string is 11.

There are 3 format specifications that do not effect the variables. Rather they are processed as they are encountered in the string. They are

Tn Tab to the n column  
Xn Space n columns  
'str' print the string in single quotes

A few words about these. If you are already at the n column or past it, then Tn will have no effect. Anything can be placed in the single quotes, except other single quotes, double quotes, and a carriage return. There are some ways around this and I'll explain them in a minute. Here is a example of using these specifiers.

```
PRINT USING "I4,' is ',H4,' in hex'",
1234,1234
```

1234 is 04CE

OK, you probably want to know where the last part went. PRINT USING processes the variables. After the second 1234, it is finished, as far as it is concerned. (Personally, I believe the format statement should be processed, using the variables as it needs them.) To get our example to print the way we wanted it to print we would have to write:

```
PRINT USING "I4,' is ',H4", 1234,1234;
PRINT " in hex"
```

This a little added work. The semicolon defeats the line feed, so the " in hex" is printed on the same line. It will look like:

1234 is 04CE in hex

Variable formats can be repeated. So "R5.2,R5.2" is the same as "2(R5.2)". This can be an handy item. The line:

```
PRINT USING "4(X2,H4)", 12,345, 7893,
9999
will print
```

000C 0159 1ED5 270F

Now what if there are more variables than format specifications. Consider the program line

```
PRINT USING "X2,H4",12,345,7893,9999
```

It will have the same output as before. The format list is reused over and over as needed. The first version is probably better since it gives you more control over the output. And it gives a better idea of what is happening.

Another feature, the variables can be mixed. Usually numbers go with number specifiers and string with string. But there are a few exceptions.

```
PRINT USING "H10","HELLO"
will print
```

48454C4C4F

This is the ASCII representation of "HELLO". The Hex Format will print almost any data type. The general rules are

H2 BYTE or BOOLEAN  
H4 INTEGER  
H10 REAL  
Hn\*2 STRING[n]

I'll let you experiment with this one on your own.

The format list is a string so it can be specified somewhere else. You could even get fancy and create an array of format statements.

```
DIM format(10):STRING(40)
format(1):="2(R8.3^)"
format(2):="T5,H5,T15,H5"
```

and so forth then when you ready to use them, use a line like:

```
PRINT USING format(1),54.3,78.2
or whatever you want.
```

I mentioned earlier that there are ways to put some of the restricted characters into the string between the single quotes. Let's print the line:

The value of "a" is 123

The double quotes cannot be used directly. But this will work:

```
a:=123
f$:="The value of "+CHR$(34)+"a"+CHR$(34)+"
is ",14"
PRINT USING f$,a
```

The CHR\$(34) puts the double quote into the string f\$. You might want to experiment with this technique. You cannot fool it with the single quote, since this only signals the start and end of a string. Experiment with PRINT USING and find new ways to improve your output.

## A DISK HIERARCHY AND INTEGRITY CHECKER

Now there is a title with a couple of big words. What it is a program to print the different levels of your disk's structure. It also reads the files looking for read errors in them. The program is in the LISTING 1 at the end of this month's column. It does make use of the PRINT USING. This program is one that I might later translate to some other language, like C. But BASIC09 makes it easier and faster to try out various things.

I left the listing in the form output by Basic09. A 4 digit number starts each line and is a running count of actual bytes used in the source. Most things take a byte for each printed character and for the EOR character. However, keywords only take a byte each. I decided to use this form, since I believe it will make spotting typographical errors a little easier. If you type a line wrong or omit something, your numbers will be different from mine. Also, notice that some lines wrap around and are continued on the next line. These will be easier to spot, because that line will not start with number. Take a look after line 03BF in the Procedure evaluate.

The main procedure is HCHECK. It prompts the user for some type of storage device and a directory on the device. After these are found out a call is made to EVALUATE, passing it the device, the pathname and level. The first level is always 0.

EVALUATE does most of the real work. It opens the current pathname as a directory for reading. Then directory entries are read one at a time. As long as they are not deleted entries or . and .., they are processed. The file descriptor sector for each is read and it is determined whether they are another directory or just a file. If a another directory is encountered, EVALUATE calls itself. The level is bumped up one. Calls are also made to DIRCHECK and FILECHECK, to test the entries readability. Also, DIRCHECK determines the number of entries in the directory.

I make use of the PRINT USING by building a string. I set up a line like:

```
f$:="T"+STR$(level*3)]
```

Actually the strings in the program are little longer. Let us say you are at level 3, that would be 4 directories in. The string would be "T9", which means indent to the 9th column. With this little trick, as we enter into different levels of directories, they are indented in the listing accordingly.

The procedure GETSECTOR gets the first 16 bytes of a files descriptor sector. The two important items we need here are the attribute information and file size. The TYPE statement in EVALUATE gives a better idea of what the first 16 bytes contain.

NAMEFIX changes the file name from system format to Basic09 format. In system format the most significant bit of the last character is set. To convert it, the bit is reset, making it a standard ASCII character. If the string is not completely filled by the name, \$FF is added to mark the end.

The procedures can be PACKed and kept in your commands directory along with RUNB, the Basic09 run time package. Then whenever you want a hierarchical listing of a directory enter HCHECK. It will ask you for a RBF device and the directory pathname. If you wish divert the output to the printer to make permanent records.

## HAVE YOU GOT YOUR TOOLS YET?

BASIC09 TOOLS are available from South East Media. They consist of 21 subprograms that I created for use with Basic09. They are all assembled object code and take less than 256 bytes, each. From time to time I will be giving you some ideas and hints on how to use them.

When writing this month's program I ran into a little problem. I wanted the procedure FILECHECK to be able to return the file size, as well as, its status. I wrote the original by reading a

byte at a time and incrementing a counter. It worked, but took a considerable amount of time. Reading all the files on a disk became a monumental task in terms of time.

My next thought was to increase the buffer size. So I changed it to 256 bytes. This speeded up things immensely. I could increase the counter by 256 bytes with each read, but the last one may have been something less than a buffer full. Unfortunately, Basic09 does not return the number of bytes with each read. So I was stuck. Read slow and count. Or speed things up and forget counting. I took the last option. To get the file size, I read the file descriptor sector, as you already know.

Actually, there was a third option. If you have the Basic09 Tools package, it contains a routine called SYSCALL. This routine lets you access any of the OS-9 system call. It literally opens the whole OS-9 system for use from Basic09. I could use \$READ. The OS-9 call \$READ works much like the Basic09 command GET. But it returns the actual count of bytes read. So, all I need do is to keep a running count of what is read on each pass.

LISTING 2 shows the alternate version of FILECHECK. This one parallels DIRCHECK; it takes a pathname and returns a count and status. To use the SYSCALL, a complex data type must be created. This is done with the data statements:

```
TYPE registers=cc,a,b,dp:BYTE; x,y,u:INTEGER
DIM reg: registers
```

The registers must be declared in this order. I reality when they are passed to SYSCALL, it sees a data area that is 10 bytes in size. So it expects the registers to be in the order declared in the TYPE statement. Only three registers are use. REG.A is the path number. REG.X is a pointer to the READ buffer. And REG.Y is the buffer size before the read and the number of bytes read after it. As a rule, they are usually the same. Here it would be 256 bytes. On the last read the Y register may be less, depending on what was left to be read.

SYSCALL does not generate any errors. Rather they are returned as part of the REG.B. I use the EXITIF...THEN...ENDEXIT construction to intercept a possible read error. If REG.CC has its 8th bit set and REG.B is not equal to 0, then some type of read error has occurred. So the status is changed to "BAD". If all goes well no errors occur. The status is returned as "GOOD" and the count is the file's size.

If you use this alternate version, you need to change the line that RUNs FILECHECK. You also can eliminate the line that computes the file's size from its file descriptor sector. In fact, you can now eliminate reading the file descriptor all together. Then you would not need to know the device on which the disk is mounted. This would make running HCHECK simpler, since it would only need to know the directory, you were checking.

You would still need to know if a name is a directory or file. The easiest way to find out if it is a directory, is to try to open it in the UPDATE+DIR mode. IF an error occurs, it is a file. Otherwise it is a directory within the directory. Use the ON ERROR GOTO... to trap the error. If no error occurs treat it as a directory. If an error #214 occurs process it as a file.

If you have any ideas of improving this program, feel free to use them. Remember the programs in the BASICALLY OS-9 column are for you to use. Change them, rewrite them, improve them or adapt them to your use. They are here for you.

#### LISTING 1

```
PROCEDURE check
0000 (* Basic: Hierarchy Check
0001 (* By: Ron Veiga
0002 (* Date: 23-DEC-88
0003 (* For the readers of Basically OS-9 column
0004 (*
0005 (* Function:
0006 (* This procedure prints a disk hierarchy indenting
0007 (* file or directory name at each level.
0008 (* Also, their size and readability are checked.
0009 (*
0010 (*
0011 (* Procedures called:
0012 (* EVALUATE, GETDEVIC, NAMEIN, DIRCHECK
0013 (* and FILECHECK.
0014 (*
0015 DIM device:STRING[32] /* device name
0016 DIM pathname:STRING[80] /* directory name
0017 DIM level:INTEGER /* indent level
0018 DIM good:BOOLEAN
0019 (* set initial level
0020 level:=0
0021 (* get device
0022 good:=FALSE
0023 REPEAT
0024 INPUT "Enter device name: ";device
0025 IF LEFT$(device,1)="/" THEN
0026 good:=TRUE
0027 ENDIF
0028 UNTIL good
0029 (* get associated path
0030 INPUT "Enter path name on device: ";pathname
0031 AOS evaluate(device,pathname,level)
0032 END
PROCEDURE evaluate
0000 (* Evaluate for use with HCHECK
0001 (*
0002 (* Passed parameters
0003 PARAM device:STRING[32]
0004 PARAM pathname:STRING[80]
0005 PARAM level:INTEGER
0006 (*
0007 (* TYPE definitions
0008 TYPE entry=NAME:STRING[20]; loc(3):BYTE
0009 TYPE sector=attr:BYTE; owner:INTEGER; lastw(5):BYTE; link:
0010 BYTE; false(4):BYTE; mode(3):BYTE
0011 (*
0012 (* Variables
0013 DIM entry /* directory entry
```

```

0019 DIM difsector \(\% Sector information
0107 DIM path:INTEGER \(\% path number
011C DIM location:REAL
0125 DIM f$:STRING(64) \(\% format string
0138 DIM count:REAL \(\% file or directory count
0160 DIM status:STRING(5) \(\% file status
017A (*
017D (* Open the directory and process it
01A1 OPEN $path,pathname:READ+DIR
01A0 WHILE NOT(EOF($path)) DO
0190 GET $path,e
01C3 FOR namefix:=name)
01C7 (* If this is a real name then get the sector information
0200 IF s.name<>" THEN
0217 location:=4096*(a.loc(1)+256*a.loc(2)+a.loc(3))
0247 NEW getsector(location,device,d)
025B (* If this is a directory then print it, GET its status
0267 (* and size, and validate it too!
0283 IF d.attr=>0 THEN
02C3 f$="*****" (level=3)+".S30," -- DIR files: ".16." Status:
".S3"
02FF NEW difsector $pathname+ "/" + s.name, count, status)
031D PRINT USING f$, s.name, count, status
0322 NEW evaluate(device, pathname+ "/" + s.name, level+1)
0332 ELSE
0336 (* Otherwise it's a file. Get its size and status.
0369 (* And print the information.
03A6 NEW filecheck (pathname+ "/" + s.name, status)
03B9 f$="*****" (level=3)+".S30," -- FILE Size: ".16." Status:
".S3"
0378 count:=d.false(1)*16777216+.d.false(2)*65536+.d.false(3)
0386 d.false(4)
042C PRINT USING f$, s.name, count, status
0431 ENDIF
0453 ENDIF
0455 ENDFILE
0459 CLOSE $path
045F END
PROCEDURE getsector
0000 (* For use with SEARCH
0010 (* Returns the first 16 bytes of a file's info sector
0040 (*
004E (* Parameters
005B PARAM loc:REAL
0062 PARAM device:STRING(32)
006E PARAM a(16):BYTE
007A (*
007D (* Variables
0089 DIM path:INTEGER
0090 OPEN $path,device+"$":READ
00A0 (* Open and read sector
00B7 SEEK $path,loc+256
00C6 GET $path,e
00D0 CLOSE $path
00D8 END
PROCEDURE namefix
0000 (* For use with SEARCH
0010 (* Converts system to Basic89 format
003A (* Name must be 29 characters long
005C (*
005F PARAM a(29):BYTE
006B DIM i:INTEGER \(\% index
007A IF a(1)=255 OR a(1)=255 OR a(1)=0 THEN
009C a(1)=255
00A7 ELSE
00AB FOR i:=1 TO 29
00B9 EXITIF a(i)>255 THEN
00C9 a(i):=a(i)-255
00D2 UNTIL i
00E2 NEXT i
00ED IF i<29 THEN
00F9 a(i+1)=255
0106 ENDIF
01A6 ENDIF
01C0 END
PROCEDURE filecheck
0000 (* For use with SEARCH
0010 (* Reads a file and returns its status
003C (*
003F (* Parameters
004C PARAM pathname:STRING(60)
0056 PARAM status:STRING(5)
0064 (*
0067 (* Variables
0073 DIM path:BYTE \REM path number
008B DIM s(256):BYTE \REM file buffer
00A2 (*
00A5 (* Error trap for bad read
00B9 ON ERROR GOTO 10
00C3 (*

```

```

00C8 (* Open the file and read it
00E4 OPEN $path,pathname:READ
00F0 status:="GOOD"
00F8 WHILE NOT(EOF($path)) DO
0106 GET $path,e
0110 ENDFILE
0114 CLOSE $path
011A END
011C (*
011F (* Error handling
0130 10 status:="BAD"
013D CLOSE $path
0143 END
PROCEDURE difsector
0000 (* For use with SEARCH
0010 (* Reads directory and counts number of entries
0045 (*
0048 (* Parameters
0055 PARAM pathname:STRING(60)
0061 PARAM count:REAL
0069 PARAM status:STRING(5)
0074 (*
0077 (* Variable
0082 DIM path:BYTE \REM path number
0087 DIM s(32):BYTE \REM directory entry buffer
009C (*
009F (* Error trap
00C0 ON ERROR GOTO 10
00C2 (*
00C5 (* Open directory and read it
00F2 OPEN $path,pathname:READ+DIR
00FE status:="GOOD"
0109 count:=0
0114 WHILE NOT(EOF($path)) DO
011F GET $path,e
0129 IF a(1)<255 AND a(1)<255 AND a(1)<0 THEN
014B count:=count+1.
015A ENDIF
015C ENDFILE
0160 CLOSE $path
0166 END
0168 (*
0169 (* Error handling
017C 10 status:="BAD"
0189 CLOSE $path
018F END
PROCEDURE filecheck
0000 (* For use with SEARCH
0010 (* Reads a file and returns its status
003C (* and its size
004B (*
004E (* Alternate version of FILECHECK
006F (* Requires Basic89 Tools' SYSCALL
0091 (*
0094 (* Parameters
00A1 PARAM pathname:STRING(60)
00AD PARAM count:REAL
00B4 PARAM status:STRING(5)
00C0 (*
00C3 (* TYPE Definitions
00D8 TYPE registers=uc,a,b,dp:BYTE a,y,u:INTEGER
00F0 (*
00FE (* Variables
010A DIM s(256):BYTE \REM file buffer
0124 DIM reg:registers \REM for call to OS9
013F (*
0142 (* Initialize the variables
0150 status:="GOOD"
0168 count:=0
0173 reg.x:=ADDR(s)
0181 reg.y:=256
018D (*
0190 (* Open the file and read it
01AC OPEN $reg,a.pathname:READ
01BC WHILE NOT(EOF($reg,e)) DO
01CA FOR sycall=(reg,409) \REM Call to ISREAD
01E9 count:=count+reg.y
01F9 EXITIF reg.count>655 AND reg.b<255 THEN
0214 status:="BAD"
021E UNTIL i
0222 ENDFILE
0226 CLOSE $reg,e
022F END
0231 (*
0234 END
ZOF

```

FOR THOSE WHO NEED TO KNOW

68 MICRO  
JOURNAL™



# Interfacing

By: Andreas Straub  
Motorola Munich  
Arabellastr. 17  
8000 Munchen 81 W. Germany

## Introduction

This is for all of you 6809 users feeling quite comfortable with your system except...floating point performance! As the recent discussion about speed improvements for fourier transformations proves, there are certain application areas where the otherwise sufficient power and speed of the 6809 gives you performance problems. The MC68881 can easily be connected to the 6809 giving you a tremendous speed improvement with floating point operations. In this article I'm going to give you all necessary information, diagrams and software examples to work successfully with this powerful and fast Floating Point Unit.

## Hardware

Like the MC68020, the MC68881 uses the so called 'dynamic bus sizing', which enables it to use data bus sizes from 8 to 32 bits. The bus size is determined by the 'SIZE'-input in combination with the 'AO' address line. Connecting the 'SIZE' pin to ground and using the 'AO' address line gives you the 8 bit bus size required for the 6809. In this case the FPU data lines D0-D7, D8-D15, D16-D23, D24-D31 have to be connected in parallel.

Before we go on two important things concerning the Vcc and Ground lines. First of all the data sheet does not show all of the Vcc and Ground pins required. (The pinout in the data book is correct, however). I include the correct pinout for reference (Fig. 1). And please remember:

***All these power lines have to be connected!***

for the device to operate properly! Secondly another warning:

***Never apply power to the MC68881 without supplying a clock***

***Heed these warnings: you run a good chance of burning out the chip. (By the way this is also true***

# the MC68881 F/P Unit (FPU) with the MC6809 CPU

for most of the other highly integrated dynamic designs like the MC68000, MC68020 and also for CPU's and support chips of other manufacturers).

So much for warnings. Now let's discuss the timing requirements of our application. Looking at the pinout we find the old favorites *RESET*, *R/5W* and *5C5S*. These are functionally identical to the equivalent 6809 signals and can therefore be directly connected to the bus. (But make sure to use *E & Q* (Pseudo VMA) in the address decoder). The *AS* signal has to be asserted while a valid address is on the bus. I decided to use the *CS* signal also for the *AS* but with a slight delay introduced by two buffers. This is necessary because both signals shouldn't be asserted at the same time. The *DS* (Data strobe) is the most difficult line to interface. First of all data has to be valid while *writing* on the *falling edge* of *DS* and on the *rising edge* while *reading*. So lets have a look at the 6809 timing diagrams (Fig. 2).

It's easy to see that the -signal would be the obvious choice for the use as DS. On the other hand we have the additional access time requirements caused by the *synchronous read cycle* while accessing the *restore & save FPU registers*. To avoid data inconsistencies between the CPU and the FPU, the FPU synchronizes the internal data to the external access when reading these registers, which gives a worst case access time of more than 300ns (12.5 MHz part). And it is exactly here that we have a problem. From the falling edge of to the rising edge of E even a 1 MHz systems gives only 250ns. For 1MHz there are certainly possible solutions not using MRDY (like using different DS-signal while reading and writing), but I have chosen the MRDY-approach because it can be used without any modifications for all CPU clockrates. (I have not excluded the other registers from the access delay because the time savings don't compensate for the additional hardware required, especially when you look at the additional software overheads).

The final circuit diagram (Fig. 3) shows the Data strobe monoflop triggered by and delayed by IC 4 giving DS as output and at the same time extending the current E-cycle using the open collector driver IC 7 connected to the MRDY input of the CPU. The delay time of IC 4 should be adjusted to 350ns (12.5 MHz FPU).

The rest of the circuit is straight forward. The FPU clock rate generated by an integrated oscillator divided by 2 gives a highly symmetrical output. The address decoding circuitry includes E & Q (Pseudo VMA) and the required address lines. The FPU uses address lines A0 to A4 giving a total memory requirement of 32 bytes. IC 3 buffers the data bus.

#### Software

Let's first have a look at the register model of the FPU (Fig. 4). For our application only 3 registers are of importance:

- the response register (16bit)
- the command register (16bit) and the
- operand register (32bit).

If we want to use the FPU with a CPU other than the MC68020 we have to emulate the firmware protocol for all data transfers by software. The basic protocol consists of the following steps:

- write the command to the command register
- wait until the response register gives you the ok to transfer data
- write your data to the operand register

- wait until the response register tells you data has been processed
- send another command to retrieve the result
- wait until the response register tells you data is ready
- read the operand register to get the result
- read the response register until you get a 'NULL'-response

In reality the total protocol is much more complicated but for our application the above minimum protocol is sufficient.

Looking at the enclosed software example you can easily find the equivalent code starting at the label 'FPU'. It is certainly a big advantage to be able to use the 16-bit 'D'-accumulator for 16bit load, store and compare operations. If anyone wants to use another CPU having only 8bit registers available, it is important to remember that the most significant byte has to be transferred first (watch out for these strange CPU's handling least significant bytes first !!!).

#### PL9 and the FPU

Because the FPU uses the standardizing IEEE floating point data format and PL9 uses a different floating point number representation a format conversion is required.

#### IEEE Floating Point format (single precision)

X	X 8 Bit X		< 23 bit >
I	I I		
sign	exponent		mantissa
bit	bias 127		unsigned
			magnitude
			(1.0 ... 2.0)

#### PL9 Floating Point Format

X	X 7 bit X	X	<23 bit >
I	I I	I	I
sign	exponent	sign	mantissa
exp	2's complement	mant	2's complement
			(0.5 ... 1.0)

The format conversion from/to IEEE format is also included in the accompanying software example. Internally the FPU always uses extended precision for all calculations (80 bit!) to deliver precise results under all conditions. (In this case the external single precision format is converted in the FPU to extended precision while transferring external data into the internal floating point registers).

### Error Handling

Basically there are two ways to find out, if an operation has been handled successfully:

- Enable exceptions and interpret the responses of the FPU

- Read and evaluate the internal FPU status register

I have chosen the second approach, because it doesn't introduce additional checking of the FPU responses for exceptions and can therefore be more easily turned on or off by the programmer. The two procedures 'checks\_on' and 'checks\_off' toggle a flag determining whether any error checking is done or not. If it's true additional code is executed reading the internal status register and checking for eventual exceptions. If an error is detected an appropriate message is displayed and a FLEX warmstart executed. This exception handler can easily be modified to suit your application specific needs.

### Speed Measurements

Finally some data regarding the speed of the Hardware/Software combination just described. Even though the speed improvements are quite remarkable, it should be noted that *most of the time is spent converting number formats*. It is therefore possible to achieve much higher ratios using all available 8 Floating point registers (I used only one) and converting data only if absolutely necessary, otherwise using register to register operations inside the FPU. I'll probably publish a fourier transformation routine using the upmost capabilities of the FPU as a PL9 ASMPROC in the near future and expect a speed improvement of more than 313030 in this case. Reason enough to build your Floating Point Hardware ?

Operation	Software	FPU	Ratio
sin()	157s	4.2s	37.4
cos()	150s	4.2s	35.7
tan()	335s	4.4s	76.1
log()	162s	4.8s	33.8

(all operations done 10000 times, 1MHz 6809, 12MHz FPU)

EOF

68-PIN GRID ARRAY

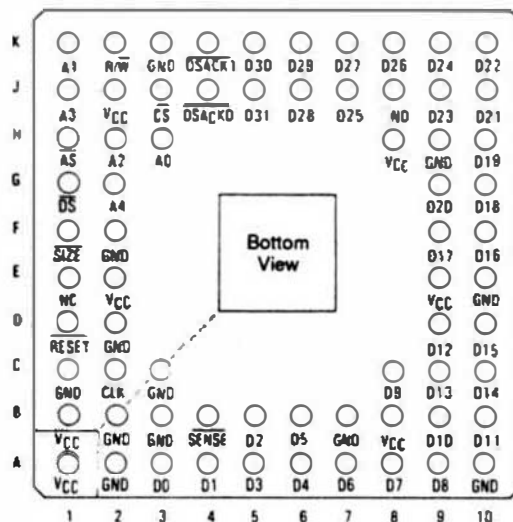


Figure 1 Correct pinout MC6809

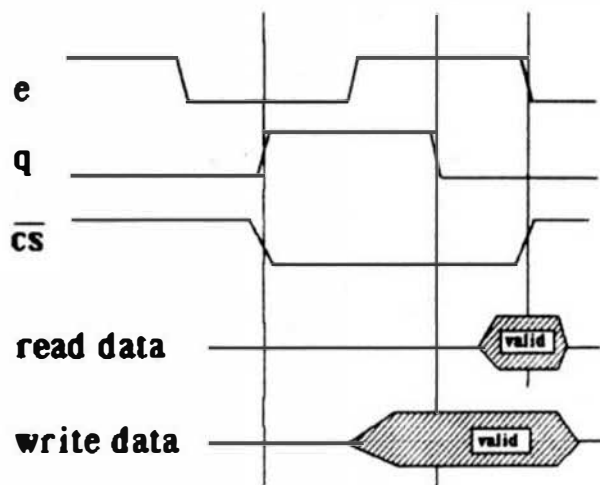


Figure 2 6809 Read/Write timing

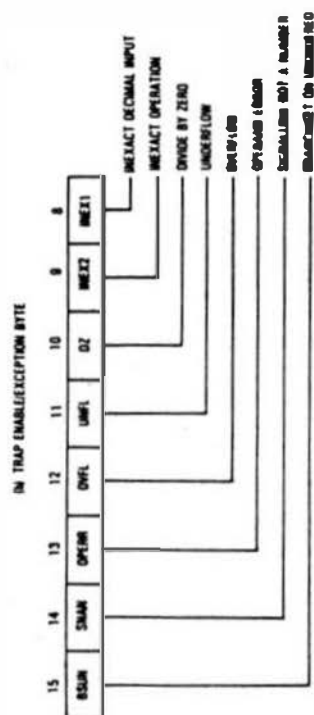
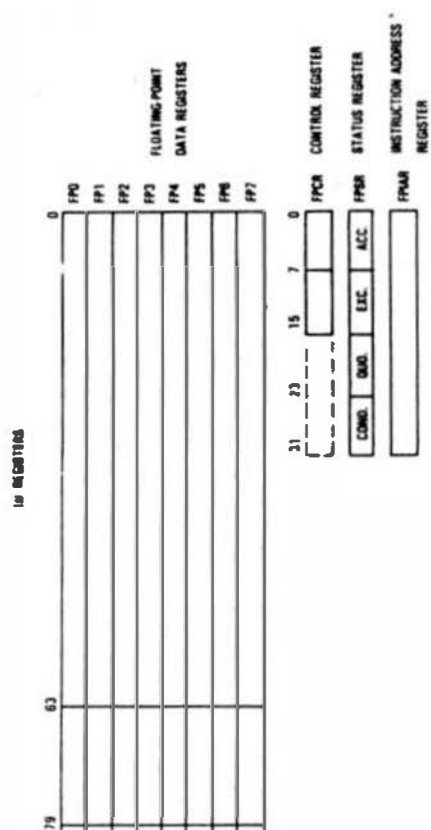


Figure 4 Register Model MC68881

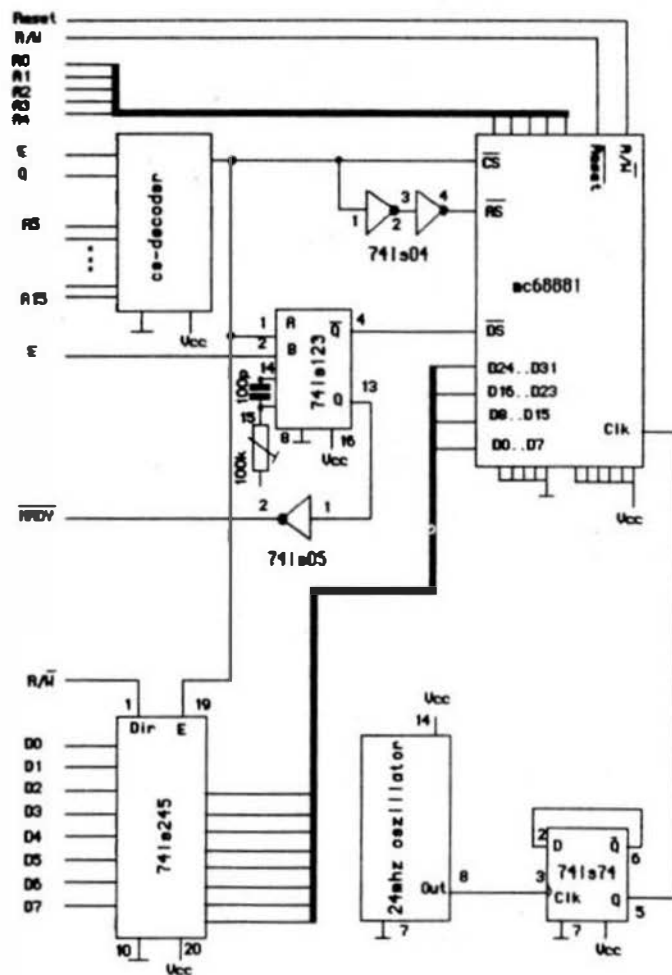


figure 3 6809-68881 interface

```
*
*      F P U . L I B
*      ~~~~~
*
*      FPU-library for the MC68001 and PL9
*          including error checks
*          position independent
*
*      To be assembled with :
*
*      MAKE, fpu.txt, G, L=fpu.lib
*
*      Andreas Straub, Munich 11/86
*
*****
*
*      FPU Equates
*
*
FFBASE      EQU      $E040
RESPONSE    EQU      FFBASE
COMMAND     EQU      FFBASE+10
OPERAND     EQU      FFBASE+16
*
*
*      FLEX EQUATES
*
*
WARM        EQU      $CD03
PUTCH       EQU      $CD18
OUTADR      EQU      $CD45
```



```

*
*  ASCII 6008TES
*
BELL      EQU    7
CR        EQU    800
LF        EQU    80A

*
*  Local variables
*

OPCODE    FDB    0
CHKFLAG   FCB    0

*****
*
*  ASMPROC enable_checks;
*
*  turn error checking on
*
*****
LDA    $3FF
STA    CHKFLAG,PCR
RTS

*****
*
*  ASMPROC disable_checks;
*
*  turn error checking off
*
*****
CLR    CHKFLAG,PCR
RTS

*****
*
*  ASMPROC mod(REAL): REAL;
*
*  Modulo remainder
*
*****
LDD    $34421
BRA    EXECFPU

*****
*
*  ASMPROC sqrt(REAL): REAL;
*
*  Square root
*
*****
LDD    $34404
BRA    EXECFPU

*****
*
*  ASMPROC ln(REAL): REAL;
*
*  logn
*
*****
LDD    $34414
BRA    EXECFPU

*****
*
*  ASMPROC log(REAL): REAL;
*
*  log10
*
*****
LDD    $34415
BRA    EXECFPU

*****
*
*  ASMPROC exp(REAL): REAL;
*
*  e**x
*
*****

```

```

LDD    $34410
BRA    EXECFPU

*****
*
*  ASMPROC alog(REAL): REAL;
*
*  10**x
*
*****
LDD    $34412
BRA    EXECFPU

*****
*
*  ASMPROC sinh(REAL): REAL;
*
*  sinh
*
*****
LDD    $34402
BRA    EXECFPU

*****
*
*  ASMPROC cosh(REAL): REAL;
*
*  cosh
*
*****
LDD    $34419
BRA    EXECFPU

*****
*
*  ASMPROC tanh(REAL): REAL;
*
*  tanh
*
*****
LDD    $34409
BRA    EXECFPU

*****
*
*  ASMPROC sin(REAL): REAL;
*
*  sine
*
*****
LDD    $34408
BRA    EXECFPU

*****
*
*  ASMPROC cos(REAL): REAL;
*
*  cosine
*
*****
LDD    $3441D
BRA    EXECFPU

*****
*
*  ASMPROC tan(REAL): REAL;
*
*  tangent
*
*****
LDD    $34407
BRA    EXECFPU

*****
*
*  ASMPROC asin(REAL): REAL;
*
*  arcsin
*
*****
LDD    $3440C
BRA    EXECFPU

```

```

*****
*
*  ASMPROC acos(REAL): REAL;
*
*  arccos
*
*****
LDD    $3441C
BRA    EXECFPU

*****
*
*  ASMPROC atan(REAL): REAL;
*
*  arctan
*
*****
LDD    $3440A

*****
*
*  Convert data on the stack
*  from 31.8 to 13.82 format
*
*****
EXECFPU  STD    >OPCODE,PCR
LDD      2.8
BNE     WOND1
LDD      4.9
BNE     NULL
WOND1    AEL     3.8
        REL     4.8
        REL     3.8
        MORA
        BPL     POSIT
        COR     3.8      2's COMPLEMENT -> SIGNED MAGNITUDE
        COR     4.8
        COR     3.8
        LDD     4.8
        ADDD    01
        STD     4.8
        LDA     3.8
        ADCA    00
        STA     3.8
        LDA     0800
POSIT    AEL     3.8
        LDB     2.8
        ADDB    0126
        LSLA
        MORA
        MCR     3.8
        STB     2.8
NULL     EQU     *

*****
*
*  Execute operation
*  result always in FPU
*
*****
FPU      LDD     >OPCODE,PCR
        STD     COMMAND
WAITR    LDD     RESPONSE
        CMPO    $0904
        BNE     WAITR
        *
        *  Write REAL-Number on the stack into FPU
        *
LOAD     LDD     2.8
        STD     OPERAND
        LDD     4.8
        STD     OPERAND+2
WAIT1    LDD     RESPONSE
        CMPO    $0902
        BNE     WAIT1
        *
        *  error checking active ?
        *  ... no continue
        *  yes ... do the checking
        *
        *  read back result from FPU
        *
        YST     >CHKFLAG,PCR
        BEQ     READY
        BSR     YSTOR

```

```

READY  LOD  $06400      MOVE FPU -> OPERAND REGISTER command
      STD  COMMAND
WAITY  LOD  RESPONSE
      CMPD  $000104      'read single precision data' response
      BNE  WAITY

```

```

*
*   Transfer data back to the stack
*

```

```

READY1 LOD  OPERAND
      STD  2.0
      LOD  OPERAND+2
      STD  4.0

```

```

WAIT2  LOD  RESPONSE      wait until FPU is finished
      CMPD  $000002
      BNE  WAIT2

```

```

*
*   Convert back from IEEE to P19 format
*

```

```

      LOD  2.0
      BNE  MCHW02
      LOD  4.0
      BQJ  BULL2
MCHW02 LOD  3.0
      AXLR
      ROL  2.0
      RORA      BIT 7 ACCU A = sign

```

```

      LOD  2.0
      STBR  $120
      STB  2.0
      LOD  3.0
      ORB  $010000000
      STB  3.0

```

```

      TSTA
      BPL  FCB
      CDB  3.0
      CDB  4.0
      CDB  5.0
      LOD  4.0
      ADDD  $1
      STD  4.0
      LOD  3.0
      ADCA  $0
      STB  3.0
      SEC
      BNA  NEXT

```

```

FCB  CLC
NEXT  RCR  3.0
      RCR  4.0
      RCR  5.0
BULL2 BCU  *

```

```

      LOD  2.0
      LOD  4.0
      RTS      put data in 'D' and 'X' registers
              (P19 convention for real return
              values)

```

```

*****
*
*   Error checking routines
*
*****

```

```

TSTOR  LOD  $0A800      read status register command
      STD  COMMAND
WAIT2  LOD  RESPONSE
      CMPD  $000104      'read single precision value'
      BNE  WAIT2
      LOD  OPERAND
      LOD  OPERAND+2      read contents, ignore upper
                          word

```

```

WAITV  LDX  RESPONSE      wait until FPU ready
      CMPL  $00002
      BNE  WAITV

```

```

AMD4  BQJ  $07C      mask out several errors
      BQJ  BERRACR      no important error occurred ...

```

```

PERR  A
      LEAX  >PERR0,PCR
      BSR  PDATA
      PULR  A

```

```

*
*   Evaluate status register and display appropriate
*   Error message in case of an exception
*

```

```

      BITA  $00100000
      BQJ  EX1
      LEAX  >BERR0,PCR

```

```

EX1  BITA  $00010000
      BQJ  EX2
      LEAX  >OPERR,PCR

```

```

EX2  BITA  $00001000
      BQJ  EX3
      LEAX  >OVERFL,PCR

```

```

EX3  BITA  $00000100
      BQJ  EX4
      LEAX  >UNDERFL,PCR

```

```

EX4  LEAX  >DIVZERO,PCR
      BSR  PDATA
      LEAX  >ADDFRAME,PCR
      BSR  PDATA
      LEAX  0.0
      JSR  OUTADR
      LEAX  >OPMES,PCR
      BSR  PDATA
      LEAX  >OPCODE,PCR
      JSR  OUTADR
      LEAX  FEEDS,PCR
      BSR  PDATA
      JSR  PDATA
      JVP  BERR0
      NORACR  RTS

```

```

*
*   Print Text without leading linefeed
*

```

```

PDATA  LOD  0.0
      CMDB  $4
      BQJ  DONE
      JSR  PUTCHR
      BNA  PDATA
      DONE  RTS

```

```

*
*   EXCEPTION MESSAGES
*

```

```

FEEDS  FCC  CR,LF,LF,BELL,4
ADDRMES FCC  / --- ADDRESS %/,4
OPMES  FCC  / --- FPU OPCODE %/,4
NANMSG  FCC  /*** NOT A NUMBER **/,4
OPERR  FCC  /*** OPERAND ERROR **/,4
OVERFL  FCC  /*** OVERFLOW **/,4
UNDERFL  FCC  /*** UNDERFLOW **/,4
DIVZERO  FCC  /*** DIVISION BY ZERO **/,4

```

EOF

FOR THOSE WHO NEED TO KNOW

68 MICRO  
JOURNAL™

# FORTH

## A Tutorial Series

By: R. D. Lurie  
9 Linda Street  
Leominster, MA 01543

### MODEM TO TEXT FILE

As I have mentioned before, I am now using a CP/M computer as a smart terminal for my GIMIX 6809, etc., system. WORDSTAR was included as part of the bundled software package, so it seemed sensible to me to find a way to use it to write programs for ultimate use on the 6809. All other considerations aside, my spelling checker is still only on the 6809, so I really needed to be able to transfer text there just to use it. Therefore, I wrote what I call WS->FLEX.

My present system consists of a simple "null-modem" connection between the terminal and the computer. The input port on the 6809 is the usual port #1 (\$E004), so that the terminal looks like a conventional input device to the computer. The current communication rate is only 2400 baud, so there is no particular pressure put on the 6809 to hurry. The 6809 is operating at 1Mhz, which I have found to be plenty fast enough for FORTH, but I could speed up to 2Mhz, if necessary. In any case, this setup was very easy to program.

Of course, this same program could be used by the 6809 to handle any sort of text input, modem or not. At 300 or 1200 baud, the 6809 would be loafing, so the only problem would be to insure that the buffer did not overflow.

### ALGORITHM

The main purpose of this utility was to move text data to a FLEX disk file in such a way that the resulting file cannot be distinguished from a similar file generated by the FLEX editor. As a result, the algorithm is rather simple.

I was able to take advantage of many of the filtering effects built into FLEX and GIMIXBUG, so the algorithm was merely to

open a FLEX disk text file, move all of the data through the normal input port into RAM, move the data from RAM to disk, and then close the file.

GIMIXBUG filters out the highest (7th) bit and FLEX accepts only printable ASCII characters, <CR>, and <HT>. This removes all of the garbage generated by WORDSTAR as print control characters, etc. The only bug is that a string of <SP>'s used to justify text will not be compressed; right now, this is not a problem.

### CONSTANTS and VARIABLES

Only one constant, one array, and one variable were used in WS->FLEX. FILE-SIZE should be set to the largest convenient size that you can manage, without overflow into the Stack area. Just pick out some arbitrary starting size for the file storage, and adjust it after you get WS->FLEX running. In my case, the longest file I ever expect to transfer would be one of these articles on FORTH for 68' MICRO JOURNAL. Therefore, the value I selected for FILE-SIZE was 25000, which is longer than I expect any single submission ever to be. In fact, I think that any file running longer than 25000 bytes should be broken up, just to protect me from Murphy's Law.

The array FILE is the place where the incoming bytes are stored until they are transferred to the disk. At present, I do not have a good way to have the computer delay further input while data are being moved to the disk, so I just have everything stop under manual control while a file is being saved. Then, again under manual control at the terminal, I start more data moving. In many cases, only one buffer-load is moved at one time, so this is not much of a problem.

The variable TALLY is a count of the bytes as they are moved. I could have kept this count on the Data Stack, but there is so little time or RAM penalty for using the variable, that I decided to go the easier route.

Unfortunately, there is no way to keep tab on how much of the file has been moved at any one time. I don't like letting a program run without any supervision, but there is no way to put any sort of information on the terminal while data are being transmitted, because the same I/O port is used at both ends of the null modem for data transfer and for terminal communication. Until I can modify the 6809 port software, there is no way that I can interrupt file transmission in order to send a response or update back to the terminal. I would appreciate any suggestions you may have on solving this problem.

### Opening the FLEX File

The first step in running the program is to open the FLEX disk file. Wilson Federici provides a convenient set of commands with FF9 for opening a FLEX file from the keyboard, while in FORTH. However, these words do not work when prompted from within a running program. Since I insist that my programs prompt for input and do not depend on me being able to remember the expected input at every place it is called for, I had to write some of my own definitions for communicating with FLEX. You may need to do the same thing, even if you are using another version of FORTH.

The definition @NAME clears the screen, homes the cursor, and prompts for the FLEX file name. EXPECT is used for fetching the input because it forces the computer to stop and wait for keyboard input. Other words, such as WORD or QUERY, do not do this, so that they are not appropriate for this kind of definition. The incoming string is stored at PAD simply by convention; and the string can never be more than 14 characters long (1+1+8+1+3). You may wonder why I used "15" in the definition if the filename could never exceed 14 characters. It is because I wanted the action of the program to be consistent, so I made sure that there would never be an automatic <CR> when the filename was being entered.

The closing phrase PAD SPAN @ sets up the string pointer and the string length on the Data Stack before exiting the routine.

FLEX-FILE completes the processing of the filename by moving it to the FCB, and reporting any error. The default extension of TXT is put in the FCB, if necessary, and the FCB is marked as being opened for writing. Then, the FLEX FMS is called to do the actual file opening, and any error condition is reported.

### Reading the MODEM Data

The data from the modem port is read from a DO ... LOOP so that it cannot overflow into the stack area of RAM. The program will stop accepting data when the buffer is full, no matter what happens at the input port. This keeps the system from crashing and preserves a large chunk of the data, even if some of it must be lost, because of a full buffer.

WS-IN first announces that the data from the port is being loaded. It then enters the loop. KEY is convenient to use to fetch the character from the port; remember that KEY does not echo the input.

There must be a printable ASCII character which is reserved to act as a flag that all of the data has been sent. Otherwise, the computer will hang in an endless loop waiting for the next character. I chose the character "accent grave" because it does not appear in English. However, you may choose any character that is appropriate. Just be sure that it will not appear in the body of any text you want to receive, because it will cause the program to stop accepting any more input data.

After checking for the terminating delimiter, the loop counter, I, is stored in TALLY. This is the easiest way to keep a running count of the bytes stored in FILE.

Finally, the value of the loop counter is added to the address at the beginning of FILE, and the byte is stored there. The routine then recycles for the next character.

The only way to escape from the input loop is to fill the buffer or to receive the terminator. Of course, you can always crash the system, but all bets are off, then!

### Move FILE to Disk

Once the reception of data is finished, the next step is to move the data in FILE to the FLEX disk file. >FLEX is used for this.

So that I will know that everything is proceeding according to plan, >FLEX tells me when it has started to save FILE. It then sets the DO ... LOOP upper limit by reading the value in TALLY. Of course, 0 is the lower loop limit. Each byte is read from FILE by adding the starting address of FILE to the value of I, the loop index, and using the sum as a pointer for C@. WRITE-BYTE is the word provided in FF9 for sending one byte to FLEX FMS. WS->FLEX WS->FLEX is the main program command which is used to invoke the program loop. It clears the buffer FILE, calls each of the subroutines, and then closes the FLEX disk file at the end. CLOSE-FILE is provided in FF9 to close the FLEX file properly after all of the data have been moved.



Incidentally, I have read that nobody is neutral on the subject of WORDSTAR, either you like it or you hate it. Well, I must admit that I do like it. I had always previously preferred an editor like the one supplied with FLEX, where everything is done with a command line. However, I have found that the arrow keys, etc., on this keyboard make WORDSTAR easy to use, so I have quickly become comfortable with it. Barring some unforeseen catastrophe, or writing an editor of my own in FORTH, I don't expect to go back to the FLEX editor, except for some trivial jobs here and there; particularly now that I can transfer text so easily to the FLEX system.

### Program Listing

The program listing is shown in the usual FLEX file format that I prefer for FF9 programs. If you use a different system, I suggest that you put each definition in a separate screen; after all, disks are now very cheap.

If you do much FLEX file I/O, you may find that @NAME, FLEX-FILE, and >FLEX would make good utility words for other programs. On the other hand, you may already have equivalent words which you can substitute into WS->FLEX. Go right ahead and do so, since these words are at the bottom of the reference chain and do not call any other routines within WS->FLEX.

### GLOSSARY

I have included a glossary (Table 1) of non-standard words to help those of you who do not use FF9 to adapt WS->FLEX to your system.

### FF9 and FF2

Wilson M. Federici, 1280 NW Grant, Corvallis, OR 97330, has been giving away a spectacularly good version of FORTH-83 for use with FLEX or SK\*DOS. He has two versions, FF9 for the 6809 and FF2 for the 6800. Send him 2 formatted disks and a postage paid mailer and he will send you, free, two versions of the binary code, a complete Assembler, and well commented source code. If you can read and follow directions, you can get FORTH running with the best system I have ever seen!

Wilson now includes directions for installing FF9 on the Color Computer, but you need a way to get a 24x80 display to take full advantage of FF9. However, it can be made to run on the 24x51 and 24x64 displays which are available for the CoCo. FF9 will NOT run without FLEX or SK\*DOS, so you cannot use Radio Shack's OS-9 instead.

I have no way to test FF2, since my 6800 system only has the old Percom disk system, but I see no reason not to expect the same great performance from FF2 that we get from FF9.

As you may know, the one thing you sacrifice when you accept free software is someone to complain to when you have a problem. Wilson Federici is very conscientious about supporting FF9/FF2, but there is a limit to what you can ask for without paying for it. You must expect to dig out of problems on your own and with the help of a few good books.

```
\ WS->FLEX CONSTANTS, etc.
25000 CONSTANT FILE-SIZE
CREATE FILE FILE-SIZE ALLOT
VARIABLE TALLY 0 TALLY !

: $NAME ( -- adr count ) \ RDL 12/26/86
  CLS
  ." FLEX file name: " \ prompt
  PAD IS EXPECT \ load FORTH input buffer
  PAD SPAN @ ; \ "adr" and "count"

: FLEX-FILE ( -- ) \ RDL 12/26/86
  $NAME \ input FLEX file name
  FCB FILENAME \ move name to FCB
  ?ERR \ report a error
  1 FCB SETEXT \ default to TXT
  2 FCB C! \ open for write
  FCB FMS \ call FLEX FMS
  ?ERR ; \ report an error

: WS-IN ( -- ) \ RDL 12/26/86
  CR ." Ready to load WORDSTAR file" CR CR
  FILE-SIZE 0 \ set loop limits
  DO KEY \ fetch character
  DUP ASCII ' ' \ - EOF
  IF LEAVE THEN \ exit reading routine
  I DUP \ count incoming bytes
  TALLY ! \ store it
  FILE + C!
  LOOP ;

: >FLEX ( -- ) \ RDL 12/26/86
  CR ." Writing FLEX file"
  TALLY @ 1+ 0 \ set DO ... LOOP limits
  ?DO FILE 1 + C! \ fetch the byte
  WRITE-BYTE \ send the byte to the disk file
  LOOP ;

: WS->FLEX ( -- ) \ RDL 12/26/86
  FILE FILE-SIZE BLANK \ clear buffer
  FLEX-FILE \ fetch name of FLEX disk file
  WS-IN \ read the WORDSTAR file
  >FLEX \ write the FLEX file
  CR ." FINISHED"
  CLOSE-FILE ;

\
CLOSE-FILE ( -- )
  Close an open disk file. Report any error.

?ERR ( -- )
  DOS error reporting routine.

FCB array
  An array of 320 bytes.

FILENAME ( string-adr count FCB -- error-code )
  Parse the counted string in the DOS buffer and move it into
  the FCB. Report any error.

FMS ( FCB -- error-code )
  Jump to DOS for all file operations except reading and
  writing a byte to the file. Report any error.

SETEXT ( extension-code FCB -- )
  Set the FLEX file extension. Default to TXT.

WRITE-BYTE ( byte -- )
  Send one byte, through the DOS, to the FCB.

Table 1: Glossary of FORTH DOS words not specifically defined in
the program WS->FLEX.
```

EOF

FOR THOSE WHO NEED TO KNOW

68 MICRO  
JOURNAL™

# RAMBLINGS

Several months back I touched briefly on the new versions of OS-9. Since then I have some additional information. Basically OS-9 for the 68000 series will be offered in three flavors. The "Personal", "Industrial" and "Professional" versions.

**Industrial OS-9™** is designed to provide system features that support critical real-time applications. The kernel is written in native assembler code for speed and minimal storage requirements. Industrial OS-9 fits into less than 24K ROM/RAM and supports over 125 function calls. Of course, like all Microware products it is ROMable.

**Personal OS-9™** is a design providing a friendly system software package. Intended for personal and educational applications, personal OS-9 combines the standard OS-9 kernel and most often used utility programs, as well as an interactive, structured BASIC.

**Professional OS-9™** is reported to be the optimum development environment for serious 68000 family systems. This version includes the Microware C compiler and several other more powerful development tools.

In addition the following development "PAKS" and other assorted development tools are now available for practically any application where critical real-time speed is a factor.

68000 development PAKS are pre-customized versions of Professional OS-9, ready to run on a variety of systems including: VME-10, MVME 101, 117, 121 and 133.

Installation PAKS are systems that provide all the modules to install OS-9/68000 or OS-9 extensions on a single computer system. Installation PAKS are provided normally for one time installation.

68000 PORTPAKS are primarily for OEMs who want to install OS-9 on their latest hardware for evaluation purposes, or on a special type system for one time application.

NETPAKS are programmer designed for those programmers desiring to install a custom OS-9 networking facility on their existing OS-9 system.

TAPEPAKS allow the installation of powerful tape backup capabilities to practically any 68000 OS-9 system.

DRIVERPAKS are designed to assist OS-9 programmers in writing and debugging OS-9 drivers and descriptors. This pak simplifies installation where specialized peripheral boards are involved.

DISKSPAKS are support programs that are intended for those who might need to install non-standard disk drivers to their existing system, such as floppy/hard disk, RAM disk, bubble memory, etc.

Microware has matured in the years we have been involved and known them (since day one). The

extent of their product support is on a par with any of the other major software vendors, including educational and custom training seminars, on-site or at the Microware facilities in Iowa.

While on the subject let me say **HAPPY BIRTHDAY!!!** to Microware. OS-9 has been a bright light in our community of users for the past 10 years. Thanks Ken and Jeanie, and all the other folks there. We really appreciate your efforts. Like they say, "You've Come a Long Way Baby!"

## Coming Features

In the near future I hope to have several new hardware and software columns going. Among them we hope to give extensive coverage to some of the other popular 68000 systems and also the VME Bus. Right now I am looking for some of you to author columns on these subjects. I would like to have some who would be willing to furnish a column every month or so. Or at least several a year. With that kind of support we can migrate into other systems and not leave any of our existing readers behind. One thing for sure, *I will not make the stupid mistake some others did by running off and hiding from our loyal USER BASE! I know who got us here. What a shame some of the others did not!*

In addition we are considering altering our size and format. We need to expand our page content. There are other important things happening that affect us all that we have not been able to present due to a lack of space. Also we need to expand our advertiser base. However, to encourage others to advertise, we have to be able to show that we have the readers that would be interested in their products, that means more of everything for all of us. One thing I will promise you, right up front. We will continue to support and report on the S50 68000(X) subjects to the extent that we now do. I will not slack off there. Like I said, *I know where we came from and who got us here!*

## Gathering

From time to time it has been suggested by many of you that we could use a real good S50 Bus & 68000 Convention, several suggested maybe this summer. Some felt that we needed to have a get-together that will allow us all a hands-on experience with some

of the new stuff available (and also important - fellowship and a good time!). I get an average of 3 or 4 calls a day from readers wanting to know if this or that is really what the ads or rumors claim.

There are hundreds and even thousands who are now running OS-9, SK-DOS, FLEX, UniFLEX, SSB DOS and other older 6800/6809 systems and want to upgrade but just don't know what to believe or think. Not only is the hardware and software an important consideration, but with things happening as they have in the past several years, many want to have some assurance, that those they spend their hard earned dollars with, have some reasonable expectations of being around awhile, with support and all that kind of stuff. The horror stories are not all tall tales!

Going to Upgrade? O.K.  
then this is maybe what  
you have been waiting for.  
A 68XXX Users Forum.

\*For All Users of \*  
OS-9, FLEX,  
SK-DOS, SSB DOS  
UniFLEX, etc.

We have had many of our readers drop by just to see some piece of hardware or software that they had heard or read about, but were not willing to make the investment without some sort of demonstration or assurance of function or quality. We have practically all of the stuff you read about here in 68 Micro Journal running in our operation, so they make a special effort to get by and try before buying. I wonder just how many more would take the effort and time to spend several days drooling over all the great new hardware/software now available.

It is really something to hear some of them as they fiddle around with one of the newer 68XXX systems or some of the better software running on the 6800/09 systems. We had a dyed-in-the-wool OS-9, level III commercial user come by just this week. He wanted to try some of his software on the Mustang020 system from Data-Comp. He dropped his 5 inch disk in, recompiled his Sculptor source programs and was running about 20 times faster in less than 15 minutes. His parting remark was that he wished that he had tried it sooner but it had seemed to be too good to be true.

We hear from readers all the time looking for this software package or another. Again they read all the stuff we print here, but it is not the same as actually doing it! *Hand-on has no substitute!* I personally answer at least a dozen calls a day from some of you wanting additional information on some item or another advertised in 68 Micro Journal (and occasionally some other magazine). I honestly try to answer all, even those that have no relationship to our group. But there is now such a wide variety of things you are interested in and we mention in 68 Micro Journal that I can no longer keep that close a tab on everything. It's not like the old days, then I could tell you a little something about most all the products that you might be interested in. Times have changed! Again, a pretty good reason to have a gathering.

If I do this, then I will want input from those of you who would be interested in attending. I need to let some of the other vendors know that you are out there and want to see his product in action. If the response is what I think it will be, then I will give the effort to make it a worthwhile project.

I feel we should have other incentives for various interests. Like sections where those with specific concerns could gather. Maybe a flea-market. Technical talks and discussions from some of the real pros. In-depth demonstrations of some of the more exotic hardware and software that is now available to us. And lots more. In fact, I would hope that the agenda would be what you expressed to me that you wanted. How about it? *Please let me know - and soon!* If you have any suggestions as to the agenda or any other part of such an undertaking, be sure to me know that also. I want it to be done right and have something of interest for all.

## MORE RAMBLINGS

This month I want to tell you a little more about the **Mustang-08™** from the Data-Comp Division of CPI. *It is a tremendous bargain, especially while they have the 1¢ sale going.* Tom said he doesn't know how long they can continue to offer a 25 megabyte hard disk (top quality popular brand) and hard disk controller in lieu of one of the standard 5 inch drives, for a penny more. But as they have always tried to bring you

top quality hardware at the very best prices, they will hold this price as long as they can. But, it may be withdrawn at any time. **Don't let this opportunity slip by!**

We have one running in our office and it is *fast!* Especially for a 68008, however, as the Mustang-08 always operates at a full 10 Mhz. (maximum for a 68008), which is faster than most others (8 Mhz.), you would expect it to be faster. Also it has no "wait states" to hold it back, as many others using inferior components must.

Ours is configured with both OS-9 and SK\*DOS 68K. Either can be booted up from the monitor menu. We partitioned the hard disk into two sections (already setup with the software and monitor provided with each system); half and half for each. This way we can run our 6809 FLEX software, 68000 cross assembled 6809 OS-9 software and, of course, all the new OS-9 and SK\*DOS 68000 software. *Now that is an unbeatable combination!* FLEX and SK\*DOS 6809 software (object) runs under SK\*DOS 68K in emulator mode without any translation. Due to the additional speed of the 68008 it is surprisingly fast. So what it boils down to is this - you simply insert your 6809 diskette into the Mustang-08 drive and run your 6809 program. Of course, if you have the source then it would be even better to re-assemble it and run as 68000 code. But for those who do not have the source, then it can still run. Meaning - you won't be down while in the changeover period! *The important thing is that the Mustang-08 is capable of running most all your 6809 FLEX/STAR\*DOS programs directly, as if it were a 6809 also.*

This explains why one of the first questions asked when a prospective buyer calls is, "Does it really run 6809 FLEX programs?" Our answer is YES! That is one of the really strong points for this system - the Mustang-08 is the **ONLY 68XXX system anywhere in the world that runs both OS-9 and FLEX type software.** I say FLEX here because most of you are familiar with that name, however, for the past several years SK\*DOS, formerly Star-DOS has been rising in popularity. And as most of you know, SK\*DOS is a greatly improved FLEX type operating system, both on 6809 systems as well as the Mustang-08. We have seen very little original 6809 FLEX software that will not perform properly on SK\*DOS systems. And there is no reason that it should not as SK\*DOS maintains all the pointers that are common to FLEX and interacts with the user in a manner similar to that of FLEX.

However, you should know that SK\*DOS is much improved over FLEX, both 6809 and 68XXX versions. Mainly due to the fact that SK\*DOS is a current product and constantly upgraded and maintained. FLEX on the other hand, has had little updating or support in the past three or four years. And most of that was done by outside sources (SWTPC, GIMIX, etc.).

(615) 842-4600  
 5900 Cassandra Smith Rd.  
 Hixson, TN 37343  
 for information  
 call (615) 842-4801  
**CoCo OS-9™ FLEX™ SOFTWARE**

# SPECIAL

## K-BASIC

K-BASIC under OS-9 and FLEX will compile TSC BASIC, XBASIC and XPC Source Code Files.

**SAVE  
\$100**

K-BASIC now makes the multitude of TSC XBASIC Software available for use under OS-9. Transfer your favorite BASIC Programs to OS-9, compile them, Assemble them, and BINGO -- useable, multi-precision, familiar Software is running under favorite Operating System!

**!!! SPECIAL ~~\$109.00~~ \$99.00 !!!**

# SCULPTOR

Full OEM & Dealer Discounts Available!

### THE SCULPTOR SYSTEM

Sculptor combines a powerful fourth generation language with an efficient database management system. Programmers currently using traditional languages such as Basic and Cobol will be amazed at what Sculptor does to their productivity. With Sculptor you'll find that what used to take a week can be achieved in just a few hours.

### AN ESTABLISHED LEADER

Sculptor was developed by professionals who needed a software development tool with capabilities that were not available in the software market. It was launched in 1981 and since then, with feedback from an ever-increasing customer base, Sculptor has been refined and enhanced to become one of the most adaptable, fast, and above all reliable systems on the market today.

### SYSTEM INDEPENDENCE

Sculptor is available on many different machines and for most operating systems, including MS DOS, Unix/Xenix and VMS. The extensive list of supported hardware ranges from small personal computers, through multi-user minis up to large minis and mainframes. Sculptor is constantly being ported to new systems.

### APPLICATION PORTABILITY

Mobility of software between different environments is one of Sculptor's major advantages. You can develop applications on a stand-alone PC and -- without any alterations to the programs -- run them on a large multi-user system. For software writers this means that their products can reach a wider marketplace than ever before. It is this system portability, together with high-speed development, that makes Sculptor so appealing to value added resellers, hardware manufacturers and software developers of all kinds.

### SPEED AND EFFICIENCY

Sculptor uses a fast and proven indexing technique which provides instant retrieval of data from even the largest of files. Sculptor's fourth generation language is compiled to a compact intermediate code which executes with impressive speed.

### INTERNATIONALLY ACCEPTED

By using a simple configuration utility, Sculptor can present information in the language and format that you require. This makes it an ideal product for software development almost anywhere in the world. Australasia, the Americas and Europe -- Sculptor is already at work in over 20 countries.

### THE PACKAGE

With every development system you receive:

- ☐ A manual that makes sense
- ☐ A periodic newsletter
- ☐ Screen form language
- ☐ Report generator
- ☐ Menu system
- ☐ Query facility
- ☐ Set of utility programs
- ☐ Sample programs

For resale products, the run-time system is available at a nominal cost.

**Facts**

**Features**

### DATA DICTIONARY

Each file may have one or more record types described. Fields may have a name, heading, type, size, format and validation list. Field type may be chosen from:

- ☐ alphanumeric
- ☐ integer
- ☐ floating point
- ☐ money
- ☐ date

### DATA FILE STRUCTURE

- ☐ Packed, fixed-length records
- ☐ Money stored in lower currency unit
- ☐ Dates stored as integer day numbers

### INDEXING TECHNIQUE

Sculptor maintains a B-tree index for each data file. Program logic allows any numbers of alternative indexes to be coded into one other file.

### INPUT DATA VALIDATION

Input data may be validated at three levels:

- ☐ automatic by field type
- ☐ validation list in data dictionary
- ☐ programmer coded logic

### ARITHMETIC OPERATORS

- Unary minus
- \* Multiplication
- / Division
- % Remainder
- + Addition
- Subtraction

### MAXIMA AND MINIMA

- Minimum key length 1 byte
- Maximum key length 160 bytes
- Minimum record length 3 bytes
- Maximum record length 32767 bytes
- Maximum fields per record 32767
- Maximum records per file 16 million
- Maximum files per program 16
- Maximum open files

Operating system limit

### PROGRAMS

- ☐ Define record layout
- ☐ Create new indexed file
- ☐ Generate standard screen-form program
- ☐ Generate standard report program
- ☐ Compile screen-form program
- ☐ Compile report program
- ☐ Screen-form program interpreter
- ☐ Report program interpreter
- ☐ Menu interpreter

### RELATIONAL OPERATORS

- = Equal to
- < Less than
- > Greater than
- <= Less than or equal to
- >= Greater than or equal to
- <> Not equal to
- and Logical and
- or Logical or
- contains Contains
- begins with Begins with

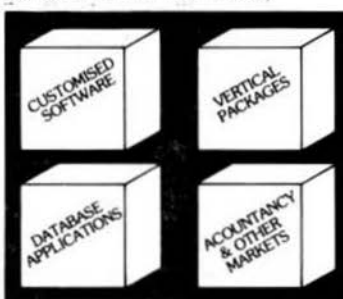
### SPECIAL FEATURES

- ☐ Full date arithmetic
- ☐ Echo suppression for passwords
- ☐ Terminal and printer independence
- ☐ Parameter passing to sub-programs
- ☐ User definable date format

### SCREEN-FORM LANGUAGE

- ☐ Query facility
- ☐ Multiform file
- ☐ Check file integrity
- ☐ Rebuild index
- ☐ Alter language and data format
- ☐ Set-up terminal characteristics
- ☐ Set-up printer characteristics
- ☐ Multiple files open in one program
- ☐ Default or programmer processing of exception conditions
- ☐ Powerful verbs for signal, display and file access
- ☐ Simultaneous display of multiple records
- ☐ Facility to call sub-programs and operating system commands
- ☐ Conditional statements
- ☐ Subroutines
- ☐ Independence of terminal type

**Sculptor for 68020  
OS-9 & UniFLEX  
\$995**



- \* Full Development Package
- \*\* Run Time Only
- \*\*\* C Key File Library

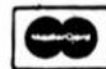
**MUSTANG-020 Users - Ask For Your Special Discount!**

	•	••	•••		•	••	•••
MUSTANG-020	\$995	\$199	\$595	IBM PC/XT/AT MSDOS	\$595	\$119	\$595
OS/9 UniFLEX	"	"	"	AT&T 3B1 UNIX	"	"	"
IBM Compatibles	"	"	"	SWTPC 68010 UniFLEX	\$1595	\$319	\$798
Tandy CoCo III	"	"	"	SWTPC 68010 UNIX	\$1990	\$398	\$995

**...Sculptor Will Run On Over 100 Other Types Of Machines...**

**...Call For Pricing...**

**!!! Please Specify Your Make of Computer and Operating System !!!**



## DISASSEMBLERS

**SUPER SLEUTH** from Computer Systems Consultants Interactive Disassembler, extremely **POWERFUL!** Disk File Binary/ASCII Examine/Change, Absolute or FULL Disassembly. XREF Generator, Label "Name Changer", and Files of "Standard Label Names" for different Operating Systems.

Color Computer SS-50 Bus (all w/ A.L. Source)  
CCD (32K Req'd) Obj. Only \$49.00  
F, \$99.00 - CCF, Obj. Only \$50.00 U, \$100.00  
CCF, w/Source \$99.00 O, \$101.00  
CCO, Obj. Only \$50.00  
OS9 68K Obj. \$100.00 w/Source \$200.00

**DYNAMITE+** - Excellent standard "Batch Mode" Disassembler. Includes XREF Generator and "Standard Label" Files. Special OS-9 options w/ OS-9 Version.

CCF, Obj. Only \$100.00 - CCO, Obj. \$ 59.95  
F, " " \$100.00 - O, object only \$150.00  
U, " " \$300.00

## PROGRAMMING LANGUAGES

**PL/9** from Windrush Micro Systems -- By Graham Trott. A combination Editor Compiler Debugger. Direct source-to-object compilation delivering fast, compact, re-entrant, ROM-able, PIC. 8 & 16-bit Integers & 6-digit Real numbers for all real-world problems. Direct control over ALL System resources, including interrupts. Comprehensive library support; simple Machine Code interface; step-by-step tracer for instant debugging. 500+ page Manual with tutorial guide.

F, CCF - \$198.00

**PASC** from S.E. Media - A Flex9 Compiler with a definite Pascal "flavor". Anyone with a bit of Pascal experience should be able to begin using PASC to good effect in short order. The PASC package comes complete with three sample programs: ED (a syntax or structure editor), EDITOR (a simple, public domain, screen editor) and CHESS (a simple chess program). The PASC package come complete with source (written in PASC) and documentation.

FLEX \$95.00

**WHIMSICAL** from S.E. MEDIA Now supports Real Numbers. "Structured Programming" WITHOUT losing the Speed and Control of Assembly Language! Single-pass Compiler features unified, user-defined I/O; produces ROMable Code; Procedures and Modules (including pre-compiled Modules); many "Types" up to 32 bit Integers, 6-digit Real Numbers, unlimited sized Arrays (vectors only); Interrupt handling; long Variable Names; Variable Initialization; Include directive; Conditional compiling; direct Code insertion; control of the Stack Pointer; etc. Run-Time subroutines inserted as called during compilation. Normally produces 10% less code than PL/9.

F and CCF - \$195.00

**KANSAS CITY BASIC** from S.E. Media - Basic for Color Computer OS-9 with many new commands and sub-functions added. A full implementation of the IF-THEN-ELSE logic is included, allowing nesting to 255 levels. Strings are supported and a subset of the usual string functions such as LEFT\$, RIGHT\$, MID\$, STRING\$, etc. are included. Variables are dynamically allocated. Also included are additional features such as Peek and Poke. A must for any Color Computer user running OS-9.

CoCo OS-9 \$39.95

**C Compiler** from Windrush Micro Systems by James McCosh. Full C for FLEX except bit-fields, including an Assembler. Requires the TSC Relocating Assembler if user desires to implement his own Libraries.

F and CCF - \$295.00

**C Compiler** from Intel - Full C except Doubles and Bit

### Availability Legends--

F - FLEX, CCF - Color Computer FLEX  
O - OS-9, CCO - Color Computer OS-9  
U - UnFLEX  
CCD - Color Computer Disk  
CCT - Color Computer Tape

\* OS-9 is a Trademark of Microware and Motorola  
\* FLEX is a Trademark of Technical Systems Consultants

Telex 5106006630

(615) 842-4600

**SOUTH EAST MEDIA**

5900 Cassandra Smith Rd.  
Hixson, TN 37343  
for information  
call (615) 842-4601

CoCo OS-9™ FLEX™

**SOFTWARE**

Fields, streamlined for the 6809. Reliable Compiler; FAST, efficient Code. More UNIX Compatible than most.

FLEX, CCF, OS-9 (Level II ONLY), U - \$575.00

**PASCAL Compiler** from Lucidata -- ISO Based P-Code Compiler. Designed especially for Microcomputer Systems. Allows linkage to Assembler Code for maximum flexibility.

F and CCF 5" - \$99.95 F 8" - \$99.95

**PASCAL Compiler** from OmegaSoft (now Certified Software) -- For the PROFESSIONAL; ISO Based, Native Code Compiler. Primarily for Real-Time and Process Control applications. Powerful; Flexible. Requires a "Motorola Compatible" Relo. Asmb. and Linking Loader.

F and CCF - \$425.00 - One Year Maint. \$100.00  
OS-9 68000 Version - \$900.00

**K BASIC** - from S.E. MEDIA -- A "Native Code" BASIC Compiler which is now Fully TSC X BASIC compatible. The compiler compiles to Assembly Language Source Code. A NEW, streamlined, Assembler is now included allowing the assembly of LARGE Compiled K-BASIC Programs. Conditional assembly reduces Run-time package.

FLEX, CCF, OS-9 Compiler / Assembler \$199.00

**CRUNCH COBOL** from S.E. MEDIA -- Supports large subset of ANSI Level 1 COBOL with many of the useful Level 2 features. Full FLEX File Structures, including Random Files and the ability to process Keyed Files. Segment and link large programs at runtime, or implemented as a set of overlays. The System requires 56K and CAN be run with a single Disk System. A very popular product.

FLEX, CCF; Normally \$199.00

Special Introductory Price \$99.95

**FORTH** from Stearns Electronics -- A CoCo FORTH Programming Language. Tailored to the CoCo! Supplied on Tape, transferable to disk. Written in FAST ML. Many CoCo functions (Graphics, Sound, etc.). Includes an Editor, Trace, etc. Provides CPU Carry Flag accessibility, Fast Task Multiplexing, Clean Interrupt Handling, etc. for the "Pro". Excellent "Learning" tool!

Color Computer ONLY - \$58.95

## DATABASE ACCOUNTING

**XDMS** from Westchester Applied Business Systems

**FOR 6809 FLEX-SK-DOS(5/8")**

Up to 32 groups/fields per record! Up to 12 character filed name! Up to 1024 byte record! User defined screen and print control! Process files! Form files! Conditional execution! Process chaining! Upward/Downward file linking! File joining! Random file virtual paging! Built in utilities! Built in text line editor! Fully session oriented! Enhanced forms! Boldface, Double width, Italics and Underline supported! Written in compact structured assembler! Integrated for FAST execution!

**XDMS-IV Data Management System**

XDMS-IV is a brand new approach to data management. It not only permits users to describe, enter and retrieve data, but

!!! Please Specify Your Operating System & Disk Size !!!

**SOUTH EAST MEDIA**

5900 Cassandra Smith Rd.  
Hixson, TN 37343  
info (615) 842-4601

CoCo OS-9™ FLEX™

**SOFTWARE**

\*\* Shipping \*\*

Add 2% U.S.A.  
(min. \$2.50)

Add 5% Surface Foreign  
10% Air Foreign





Tele 5108008630  
(615) 842-4600

**SOUTH EAST  
MEDIA**

5900 Cassandra Smith Rd.  
Hixson, TN 37343  
for information  
call (615) 842-4601

CoCo OS-9™ FLEX™  
**SOFTWARE**

also to process entire files producing customized reports, screen displays and file output. Processing can consist of any of a set of standard high level functions including record and field selection, sorting and aggregation, lookups in other files, special processing of record subsets, custom report formatting, totaling and subtotaling, and presentation of up to three related files as a "database" on user defined output reports.

#### POWERFUL COMMANDS

XDMS-IV combines the functionality of many popular DBMS software systems with a new easy to use command set into a single integrated package. We've included many new features and commands including a set of general file utilities. The processing commands are Input-Process-Output (IPO) oriented which allows almost instant implementation of a process design.

#### SESSION ORIENTED

XDMS-IV is session oriented. Enter "XDMS" and you are in instant command of all the features. No more waiting for a command to load in from disk! Many commands are immediate, such as CREATE (file definition), UPDATE (file editor), PURGE and DELETE (utilities). Others are process commands which are used to create a user process which is executed with a RUN command. Either may be entered into a "process" file which is executed by an EXECUTE statement. Processes may execute other processes, or themselves, either conditionally or unconditionally. Menus and screen prompts are easily coded, and entire user applications can be run without ever leaving XDMS-IV!

#### IT'S EASY TO USE

XDMS-IV keeps data management simple! Rather than design a complex DBMS which hides the true nature of the data, we kept XDMS-IV file oriented. The user view of data relationships is presented in reports and screen output, while the actual data resides in easy to maintain files. This aspect permits customized presentation and reports without complex redefinition of the database files and structure. XDMS-IV may be used for a wide range of applications from simple record management systems (addresses, inventory ...) to integrated database systems (order entry, accounting...). The possibilities are unlimited...

FOR 6809 FLEX-SK-DOS(5/8")

\$249.95

## ASSEMBLERS

ASTRUK09 from S.E. Media -- A "Structured Assembler for the 6809" which requires the TSC Macro Assembler.  
F, CCF, \$99.95

Macro Assembler for TSC -- The FLEX STANDARD Assembler.

Special -- CCF \$35.00; F \$50.00

OSM Extended 6809 Macro Assembler from Lloyd I/O. -- Provides local labels, Motorola S-records, and Intel Hex records; XREF. Generate OS-9 Memory modules under FLEX.

FLEX, CCF, OS-9 \$99.00

Relocating Assembler/Linking Loader from TSC. -- Use with many of the C and Pascal Compilers.

F, CCF \$150.00

MACE, by Graham Trott from Windrush Micro Systems -- Co-Resident Editor and Assembler; fast interactive A.L. Programming for small to medium-sized Programs.

F, CCF - \$75.00

XMACB -- MACE w/Cross Assembler for

68

00/1/2/3/8 F, CCF - \$98.00

## CROSS ASSEMBLERS

TRUE CROSS ASSEMBLERS from Computer Systems Consultants -- Supports 1802/5, Z-80, 6800/1/2/3/8/11/HC11, 6804, 6805/HC05/ 146805, 6809/00/01, 6502 family, 8080/5, 8020/1/2/3/5/C35/39/40/48/C48/49/C49/50/8748/49, 8031/51/8751, and 68000 Systems. Assembler and Listing formats same as target CPU's format. Produces machine independent Motorola S-Text.

68000 or 6809, FLEX, CCF, OS-9, UniFLEX

any object or source each - \$50.00

any 3 object or source each - \$100.00

Set of ALL object \$200.00 - w/source \$500.00

XASM Cross Assemblers for FLEX from S.E. MEDIA -- This set of 6800/1/2/3/5/8, 6301, 6502, 8080/5, and Z80 Cross Assemblers uses the familiar TSC Macro Assembler Command Line and Source Code format, Assembler options, etc., in providing code for the target CPU's.

Complete set, FLEX only - \$150.00

CRASMB from LLOYD I/O -- Supports Motorola's, Intel's, Zilog's, and other's CPU syntax for these 8-Bit microprocessors: 6800, 6801, 6303, 6804, 6805, 6809, 6811 (all varieties); 6502, 1802/5, 8048 family, 8051 family, 8080/85, Z8, Z80, and TMS-7000 family. Has MACROS, Local Labels, Label X-REF, Label Length to 30 Chars. Object code formats: Motorola S-Records (text), Intel HEX-Records (text), OS9 (binary), and FLEX (binary). Written in Assembler ... e.g. Very Fast.

CPU TYPE - Price each:

For:	MOTOROLA	INTEL	OTHER	COMPLETE SET
FLEX9	\$150	\$150	\$150	\$399
OS9/6809	\$150	\$150	\$150	\$399
OS9/68K	-----	-----	-----	\$432

CRASMB 16.32 from LLOYD I/O -- Supports Motorola's 68000, and has same features as the 8 bit version. OS9/68K Object code Format allows this cross assembler to be used in developing your programs for OS9/68K on your OS9/6809 computer.

FLEX, CCF, OS-9/6809 \$249.00

## UTILITIES

Basic09 XRef from S.E. Media -- This Basic09 Cross Reference Utility is a Basic09 Program which will produce a "pretty printed" listing with each line numbered, followed by a complete cross referenced listing of all variables, external procedures, and line numbers called. Also includes a Program List Utility which outputs a fast "pretty printed" listing with line numbers. Requires Basic09 or RunB.

O & CCO obj. only -- \$39.95; w/ Source - \$79.95

BTree Routines - Complete set of routines to allow simple implementation of keyed files - for your programs - running under Basic09. A real time saver and should be a part of every serious programmers tool-box.

O & CCO obj. only - \$89.95

Lucidata PASCAL UTILITIES (Requires LUCIDATA Pascal ver 3)

XREF -- produce a Cross Reference Listing of any text; oriented to Pascal Source.

INCLUDE -- Include other Files in a Source Text, including Binary - unlimited nesting.

PROFILER -- provides an Indented, Numbered, "Strucogram" of a Pascal Source Text File; view the overall structure of large programs, program integrity, etc. Supplied in Pascal Source Code; requires compilation.

F, CCF --- EACH 5" - \$40.00, 8" - \$50.00

DUB from S.E. Media -- A UniFLEX BASIC decompiler Re-Create a Source Listing from UniFLEX Compiled basic Programs. Works w/ ALL Versions of 6809 UniFLEX basic. U - \$219.95

**LOW COST PROGRAM KITS from Southeast Media** -- The following kits are available for FLEX on either 5 or 8 inch disk.

1. **BASIC TOOL-CHEST \$29.95**  
BLISTER.CMD: pretty printer  
LINEXREF.BAS: line cross-referencer  
REMPAC.BAS, SPCPAC.BAS, COMPAC.BAS: remove superfluous code  
STRIP.BAS: superfluous line-numbers stripper
2. **FLEX UTILITIES KIT \$39.95**  
CATS.CMD: alphabetically-sorted directory listing  
CATD.CMD: date-sorted directory listing  
COPYSORT.CMD: file copy, alphabetically  
COPYDATE.CMD: file copy, by date-order  
FILEDATE.CMD: change file creation date  
INFO.CMD (& INFOGMX.CMD): tells disk attributes & contents  
RELINK.CMD (& RELINK82): re-orders fragmented free chain
3. **ASSEMBLERS/DISASSEMBLERS UTILITIES \$39.95**  
LINEFBED.CMD: 'modularise' disassembler output  
MATH.CMD: decimal, hex, binary, octal conversions & tables  
SKIP.CMD: column stripper
4. **WORD - PROCESSOR SUPPORT UTILITIES \$49.95**  
FULLSTOP.CMD: checks for capitalization where required  
BSTYCTT.BAS (.BAC): Stylo to dot-matrix printer program  
NECPRINT.CMD: Stylo to dot-matrix printer filter code
5. **UTILITIES FOR INDEXING \$49.95**  
MENU.BAS: selects required program from list below  
INDEX.BAC: word index  
PIRASES.BAC: phrase index  
CONTENT.BAC: table of contents  
INDXSORT.BAC: fast alphabetic sort routine  
FORMATER.BAC: produces a 2-column formatted index  
APPEND.BAC: append any number of files  
CHAR.BIN: line reader

**FULL SCREEN FORMS DISPLAY** from Computer Systems Consultants -- TSC Extended BASIC program supports any Serial Terminal with Cursor Control or Memory-Mapped Video Displays; substantially extends the capabilities of the Program Designer by providing a table-driven method of describing and using Full Screen Displays.

*F and CCF, U - \$25.00, w/ Source - \$50.00*

**SOLVE** from S.E. Media - OS-9 Levels I and II only. A Symbolic Object/Logic Verification & Examine debugger. Including inline debugging, disassemble and assemble. SOLVE IS THE MOST COMPLETE DEBUGGER we have seen for the 6809 OS-9 series! SOLVE does it all! With a rich selection of monitor, assembler, disassembler, environmental, execution and other miscellaneous commands, SOLVE is the MOST POWERFUL tool-kit item you can own! Yet, SOLVE is simple to use! With complete documentation, a snap! Everyone who has ordered this package has raved! See review - 68 Micro Journal - December 1985. No 'blind' debugging here, full screen displays, rich and complete in information presented. Since review in 68 Micro Journal, this is our fastest mover!

*Levels I & II only - OS-9 Regular \$149.95  
SPECIAL INTRODUCTION OFFER \$69.95*

#### Availability Legends--

F = FLEX, CCF = Color Computer FLEX  
O = OS-9, CCO = Color Computer OS-9  
U = Uniflex  
CCD = Color Computer Disk  
CCT = Color Computer Tape

\* OS-9 is a Trademark of Microvare and Motorola  
\* FLEX is a Trademark of Technical Systems Consultants



## DISK UTILITIES

**OS-9 VDisk** from S.E. Media -- For Level I only. Use the Extended Memory capability of your SWTPC or Gimix CPU card (or similar format DAT) for FAST Program Compiles, CMD execution, high speed inter-process communications (without pipe buffers), etc. - SAVE that System Memory. Virtual Disk size is variable in 4K increments up to 960K. Some Assembly Required.

*Level I OS-9 obj. \$79.95; w/ Source \$149.95*

**O-F** from S.E. Media -- Written in BASIC09 (with Source), includes: REFORMAT, a BASIC09 Program that reformats a chosen amount of an OS-9 disk to FLEX Format so it can be used normally by FLEX; and FLEX, a BASIC09 Program that does the actual read or write function to the special O-F Transfer Disk; user-friendly menu driven. Read the FLEX Directory, Delete FLEX Files, Copy both directions, etc. FLEX users use the special disk just like any other FLEX disk

*O - 6809/68000 \$79.95*

**LSORT** from S.E. Media - A SORT/MERGE package for OS-9 (Level I & II only). Sorts records with fixed lengths or variable lengths. Allows for either ascending or descending sort. Sorting can be done in either ASCII sequence or alternate collating sequence. Right, left or no justification of data fields available. LSORT includes a full set of comments and error messages.

*OS-9 \$85.00*

**HIER** from S.E. Media - HIER is a modern hierarchical storage system for users under FLEX. It answers the needs of those who have hard disk capabilities on their systems, or many files on one disk - any size. Using HIER a regular (any) FLEX disk (8 - 5 - hard disk) can have sub-directories. By this method the problems of assigning unique names to files is less burdensome. Different files with the exact same name may be on the same disk, as long as they are in different directories. For the winchester user this becomes a must. Sub-directories are the modern day solution that all current large systems use. Each directory looks to FLEX like a regular file, except they have the extension '.DIR'. A full set of directory handling programs are included, making the operation of HIER simple and straightforward. A special install package is included to install HIER to your particular version of FLEX. Some assembly required. Install indicates each byte or reference change needed. Typically - 6 byte changes in source (furnished) and one assembly of HIER is all that is required. No programming required!

*FLEX - SK\*DOS \$79.95*

!!! Please Specify Your Operating System & Disk Size !!!



\*\* Shipping \*\*

Add 2% U.S.A.  
(min. \$2.50)  
Add 5% Surface Foreign  
10% Air Foreign



Tel: 5108006830  
**(615) 842-4600**

**SOUTH EAST  
MEDIA**

5900 Cassandra Smith Rd.  
Hixson, TN 37343  
for information  
call (615) 842-4600

CoCo OS-9™ FLEX™  
**SOFTWARE**

**COPYMULT** from S.E. Media - Copy LARGE Disks to several smaller disks. FLEX utilities allow the backup of ANY size disk to any SMALLER size diskettes (Hard Disk to floppies, 8" to 5", etc.) by simply inserting diskettes as requested by COPYMULT. No fooling with directory deletions, etc. COPYMULT.CMD understands normal "copy" syntax and keeps up with files copied by maintaining directories for both host and receiving disk system. Also includes BACKUP.CMD to download any size "random" type file; RESTORB.CMD to restructure copied "random" files for copying, or recopying back to the host system; and FREELINK.CMD as a "bonus" utility that "relinks" the free chain of floppy or hard disk, eliminating fragmentation.

Completely documented Assembly Language Source files included.

ALL 4 Programs (FLEX, 8" or 5") \$99.50

**COPYCAT** from Lucidata -- Pascal NOT required. Allows reading TSC Mini-FLEX, SSB DOS68, and Digital Research CP/M Disks while operating under FLEX 1.0, PLEX 2.0, or FLEX 9.0 with 6800 or 6809 Systems. COPYCAT will not perform miracles, but, between the program and the manual, you stand a good chance of accomplishing a transfer. Also includes some Utilities to help out. Programs supplied in Modular Source Code (Assembly Language) to help solve unusual problems.

F and CCF 5" - \$50.00 F 8" - \$65.00

**VIRTUAL TERMINAL** from S.E. Media - Allows one terminal to do the work of several. The user may start as many as eight task on one terminal, under VIRTUAL TERMINAL and switch back and forth between task at will. No need to exit each one; just jump back and forth. Complete with configuration program. The best way to keep up with those background programs.

O & CCO - obj. only - \$49.95

**FLEX DISK UTILITIES** from Computer Systems Consultants - Eight (8) different Assembly Language (w/ Source Code) FLEX Utilities for every FLEX Users Toolbox: Copy a File with CRC Errors; Test Disk for errors; Compare two Disks; a Fast Disk Backup Program; Edit Disk Sectors; Linearize Free-Chain on the Disk; print Disk Identification; and Sort and Replace the Disk Directory (in sorted order). -- PLUS -- Ten XBASIC Programs including: A BASIC Reorganizer with EXTRAs over "RENUM" like check for missing label definitions, processes Disk to Disk instead of in Memory, etc. Other programs Compare, Merge, or Generate Updates between two BASIC Programs, check BASIC Sequence Numbers, compare two unsequenced files, and 5 Programs for establishing a Master Directory of several Disks, and sorting, selecting, updating, and printing paginated listings of these files. A BASIC Cross-Reference Program, written in Assembly Language, which provides an X-Ref Listing of the Variables and Reserved Words in TSC BASIC, XBASIC, and PRECOMPILER BASIC Programs.

ALL Utilities include Source2 (either BASIC or A.L. Source Code).

F and CCF - \$50.00

BASIC Utilities ONLY for UniFLEX - \$30.00

## GAMES

**RAPIER** - 6809 Chess Program from S.E. Media -- Requires FLEX and Displays on Any Type Terminal. Features: Four levels of play. Swap side. Point scoring system. Two display boards. Change skill level. Solve Checkmate problems in 1-2-3-4 moves. Make move and swap sides.

Play white or black. This is one of the strongest CHESS programs running on any microcomputer, estimated USCF Rating 1600+ (better than most 'club' players at higher levels)

F and CCF - \$79.95

## COMMUNICATIONS

**CMODEM** Telecommunications Program from Computer Systems Consultants, Inc. -- Menu-Driven; supports Dumb-Terminal Mode, Upload and Download in non-protocol mode, and the CP/M "Modem7" Christensen protocol mode to enable communication capabilities for almost any requirement. Written in "C".

FLEX, CCF, OS-9, UniFLEX, 68000 & 6809 w/ th Source \$100.00 - without Source \$50.00

**X-TALK** from S.E. Media - X-TALK consists of two disks and a special cable, the hookup enables a 6809 SWTPC computer to dump UniFLEX files directly to the UniFLEX MUSTANG-020. This is the ONLY currently available method to transfer SWTPC 6809 UniFLEX files to a 68000 UniFLEX system. Gimix 6809 users may dump a 6809 UniFLEX file to a 6809 UniFLEX five inch disk and it is readable by the MUSTANG-020. The cable is specially prepared with internal connections to match the non-standard SWTPC SO/9 I/O Db25 connectors. A special SWTPC S+ cable set is also available. Users should specify which SWTPC system he/she wishes to communicate with the MUSTANG-020. The X-TALK software is furnished on two disks. One eight inch disk contains S.E. Media modem program C-MODEM (6809) and the other disk is a MUSTANG-020 five inch disk with C-MODEM (68020). Text and binary files may be directly transferred between the two systems. The C-MODEM programs are unaltered and perform as excellent modem programs also. X-TALK can be purchased with or without the special cables, but this special price is available to registered MUSTANG-020 users only.

X-TALK Complete (cable, 2 disks) \$99.95

X-TALK Software (2 disks only) \$69.95

X-TALK with C-MODEM Source \$149.95

**XDATA** from S.E. Media - A COMMUNICATION Package for the UniFLEX Operating System. Use with CP/M, Main Frames, other UniFLEX Systems, etc. Verifies Transmission using checksum or CRC; Re-Transmits bad blocks, etc.

U - \$299.99

## EDITORS & WORD PROCESSING

**JUST** from S.E. Media -- Text Formatter developed by Ron Anderson; for Dot Matrix Printers, provides many unique features. Output "Formatted" Text to the Display. Use the FPRINT.CMD supplied for producing multiple copies of the "Formatted" Text on the Printer INCLUDING IMBEDDED PRINTER COMMANDS (very useful at other times also, and worth the price of the program by itself). "User Configurable" for adapting to other Printers (comes set up for Epson MX-80 with Graftmax); up to ten (10) imbedded "Printer Control Commands". Compensates for a "Double Width" printed line. Includes the normal line width, margin, indent, paragraph, space, vertical skip lines, page length, page numbering, centering, fill, justification, etc. Use with PAT or any other editor.

\* Now supplied as a two disk set:

Disk #1: JUST2.CMD object file, JUST2.TXT PL9 source: FLEX - CC

Disk #2: JUSTSC object and source in C: FLEX - OS9 - CC

The JTSC and regular JUST C source are two separate programs. JTSC compiles to a version that expects TSC Word Processor type commands, (.pp .sp .oc etc.) Great for your older text files. The C source compiles to a standard syntax JUST.CMD object file. Using JUST syntax (.p .u .y etc.) With all JUST functions plus several additional printer formatting functions. Reference the JUSTSC C source. For those wanting an excellent BUDGET PRICED word processor, with features none of the others have. This is it!

Disk (1) - PL9 FLEX only - F & CCF - \$49.95

Disk Set (2) - F & CCF & OS9 (C version) - \$69.95

OS-9 68K000 complete with Source - \$79.95

PAT from S.E. Media - A full feature screen oriented TEXT EDITOR with all the best of "PIE". For those who swore by and loved only FIB, this is for you! All FIB features and much more! Too many features to list. And if you don't like these, change or add your own. PL-9 source furnished. "C" source available soon. Easily configured to your CRT, with special config section.

Regular FLEX \$129.50  
\* SPECIAL INTRODUCTION OFFER \* \$79.95

SPECIAL PAT/JUST COMBO (w/source)  
FLEX \$99.95

OS-9 68K Version \$229.00

SPECIAL PAT/JUST COMBO 68K \$249.00

Note: JUST in "C" source available for OS-9

CEDRIC from S.E. Media - A screen oriented TEXT EDITOR with availability of 'MENU' aid. Macro definitions, configurable 'permanent definable MACROS' - all standard features and the fastest 'global' functions in the west. A simple, automatic terminal config program makes this a real 'no hassle' product. Only 6K in size, leaving the average system over 165 sectors for text buffer - approx. 14,000 plus of free memory! Extra fine for programming as well as text.

Regular \$129.95

SPECIAL INTRODUCTION OFFER FLEX \$69.95

BAS-EDIT from S.E. Media - A TSC BASIC or X BASIC screen editor. Appended to BASIC or X BASIC, BAS-EDIT is transparent to normal BASIC/X BASIC operation. Allows editing while in BASIC/X BASIC. Supports the following functions: OVERLAY, INSERT and DUP LINE. Make editing BASIC/X BASIC programs SIMPLE! A GREAT time and effort saver. Programmers love it! NO more retyping entire lines, etc. Complete with over 25 different CRT terminal configuration overlays.

FLEX, CCF, STAR-DOS Regular \$69.95

Limited Special Offer: \$39.95

SCREDITOR III from Windrush Micro Systems - Powerful Screen-Oriented Editor/Word Processor. Almost 50 different commands; over 300 pages of Documentation with Tutorial. Features Multi-Column display and editing, "decimal align" columns (AND add them up automatically), multiple keystroke macros, even/odd page headers and footers, imbedded printer control codes, all justifications, "help" support, store common command series on disk, etc. Use supplied "set-up", or remap the keyboard to your needs. Except for proportional printing, this package will DO IT ALL!

6800 or 6809 FLEX or SSB DOS, OS-9 - \$175.00

SPELLB "Computer Dictionary" from S.E. Media - OVER 150,000 words! Look up a word from within your Editor or Word Processor (with the SPH.CMD Utility which operates in the FLEX UCS). Or check and update the Text after entry; ADD WORDS to the Dictionary, "Flag" questionable words in the Text, "View a word in context" before changing or ignoring, etc. SPELLB first checks a "Common Word Dictionary", then the normal Dictionary, then a "Personal Word List", and finally, any "Special Word List" you may have specified. SPELLB also allows the use of Small Disk Storage systems.

F and CCF - \$129.95

STYLO-GRAPH from Great Plains Computer Co. - A full-screen oriented WORD PROCESSOR -- (uses the 51 x 24 Display Screens on CoCo PLEX/STAR-DOS, or PBJ Wordpak). Full screen display and editing; supports the Daisy Wheel proportional printers.

NEW PRICES 6809 CCF and CCO - \$99.95,

F or O - \$179.95, U - \$299.95

STYLO-SPELL from Great Plains Computer Co. - Fast Computer Dictionary. Complements Stylograph.

NEW PRICES 6809 CCF and CCO - \$69.95,

F or O - \$99.95, U - \$149.95

#### Availability Legends--

F - FLEX, CCF - Color Computer FLEX  
O - OS-9, CCO - Color Computer OS-9  
U - Uniflex  
CCD - Color Computer Disk  
CCT - Color Computer Tape

\* OS-9 is a Trademark of Microware and Motorola  
\* FLEX is a Trademark of Technical Systems Consultants



STYLO-MERGE from Great Plains Computer Co. - Merge Mailing List to "Form" Letters, Print multiple Files, etc., through Stylo.

NEW PRICES 6809 CCF and CCO - \$59.95,

F or O - \$79.95, U - \$129.95

STYLO-PAK --- Graph + Spell + Merge Package Deal!!!

F or O - \$329.95, U - \$549.95

O, 68000 \$595.00

## MISCELLANEOUS

TABULA RASA SPREADSHEET from Computer Systems Consultants - TABULA RASA is similar to DESKTOP/PLAN; provides use of tabular computation schemes used for analysis of business, sales, and economic conditions. Menu-driven; extensive report-generation capabilities. Requires TSC's Extended BASIC.

F and CCF, U - \$50.00, w/ Source - \$100.00

DYNACALC - Electronic Spread Sheet for the 6809 and 68000.

F, OS-9 and SPECIAL CCF - \$200.00, U - \$395.00

OS-9 68K - \$595.00

FULL SCREEN INVENTORY/MRP from Computer Systems Consultants - Use the Full Screen Inventory System/Materials Requirement Planning for maintaining inventories. Keeps item field file in alphabetical order for easier inquiry. Locate and/or print records matching partial or complete item, description, vendor, or attributes; find backorder or below stock levels. Print-outs in item or vendor order. MRP capability for the maintenance and analysis of Hierarchical assemblies of items in the inventory file. Requires TSC's Extended BASIC.

F and CCF, U - \$50.00, w/ Source - \$100.00

FULL SCREEN MAILING LIST from Computer Systems Consultants - The Full Screen Mailing List System provides a means of maintaining simple mailing lists. Locate all records matching on partial or complete name, city, state, zip, or attributes for Listings or Labels, etc. Requires TSC's Extended BASIC.

F and CCF, U - \$50.00, w/ Source - \$100.00

DIET-TRAC Forecaster from S.E. Media - An X BASIC program that plans a diet in terms of either calories and percentage of carbohydrates, proteins and fats (C P G%) or grams of Carbohydrate, Protein and Fat food exchanges of each of the six basic food groups (vegetable, bread, meat, skim milk, fruit and fat) for a specific individual. Sex, Age, Height, Present Weight, Frame Size, Activity Level and Basal Metabolic Rate for normal individual are taken into account. Ideal weight and sustaining calories for any weight of the above individual are calculated. Provides number of days and daily calendar after weight goal and calorie plan is determined.

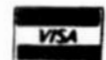
F - \$59.95, U - \$89.95

!!! Please Specify Your Operating System & Disk Size !!!



\*\* Shipping \*\*

Add 2% U.S.A.  
(min. \$2.50)  
Add 5% Surface Foreign  
10% Air Foreign



SK\*DOS is currently a single user, single tasking product. However, I talked just this week with several very capable software engineers who are working on several HLLs and multi-tasking, multi-user functions for SK\*DOS, as well as a hierarchal disk filing system. This speaks well for the future of SK\*DOS. With both OS-9 and SK\*DOS you have a full range of choices. *And the clincher is that you can purchase a Mustang-08 with both operating systems for nearly a thousand dollars (\$1,000.00) less than the next best thing. Think about that: more RAM, speed, versatility and quality support than any other 68XXX system available, for a cool \$1,000.00 or so less!* But, like I said previously, I don't know how long this reduced price can last.

The Mustang-08 comes standard with 768K of usable RAM installed. That is the most you can use, considering the 68008 only addresses 1 megabyte of RAM. Most of the remainder of addressable memory space is dedicated to system ROM/EPROM, etc.

At this point let me say just a little more about RAM. Plainly put - "the more the better!" It has been our experience that with 68XXX systems 128K is practically useless. Fact is, most systems need at the very, very least 128K just to start to do anything. Just look at the size of most commands written in any HLL, they alone will take up most of 128K. Then there is the next step up, 256K, which is probably the very least you can have available and get anything worthwhile done. Granted many offer you the option to upgrade (at additional expense) to 512K maximum. Unlike most, the Mustang-08 comes standard with 768K of "no wait state" RAM. With this we set up a 400K or so RAM disk, and the system archives back to the hard disk all the new work we do about every ten minutes, in background, so the regular operation never knows about it, or cares. This is initiated from a procedure file that looks at the clock periodically and updates all files that have changed in time since the last automatic update period. Thus we will lose very little if any data should any sort of "flake out" occur. The advantage of RAM disk operation is that we load all the CMD files needed, as well as any other files we will be using, to RAM disk (about 40 seconds), from a pre-written procedure file. The speed advantage over disk/RAM operations is about 20 to one. A considerable factor of efficiency gain.

So if you only have 128-512K of RAM to start, you may not (most times will not) have sufficient RAM space to get all the CMDS, source and other data you need into your RAM disk. The more RAM you have, the more efficient your operation will be. And in some cases it will be practically impossible to do some operations without the maximum amount of RAM. Don't skimp on this point. It may be the most important consideration of all.

Also the onboard time-of-day clock, which

can be battery backed up is a very important consideration. Not only for convenience at boot time, but for many other reasons too numerous to go into here. Sufficient to say that the clock is an essential part of the total operation, especially with battery backup.

Benchmarks are sorta like clouds, they can take all kinds of shapes and sizes. That is why, when we had our Mustang-020 benchmarked, we used industry standard test procedures and algorithms done by an independent source (see benchmarks, both ads, this issue). We wanted them to be believed because they are important to both you as a prospective buyer and us to qualify our claims of quality and efficiency. Even the prestigious journal "Dr. Dobbs Journal" rated the Mustang-020 one of the very best. They referred to it as "The Workhorse" (February 1987 issue). *We have developed the Mustang-08 in the same atmosphere of quality control!*

Considering the price, and inclusion of either OS-9 Level II or SK\*DOS, this is a system that is hard to pass up. And it comes **complete ready to run!** Just so there is no misunderstanding, let me repeat that we include level II of OS-9 68K, which is the very latest version out. As for SK\*DOS, it is the very latest also. In addition we will offer a very liberal update policy for SK\*DOS. We keep on trying!

## An OS-9 Archive Program

One type of program that we are getting a lot of calls for is an OS-9 Archive-BackUp Program. Both for the 6809 and 68XXX. So we commissioned a programmer to write one.

It should be available by the time you read this, but these specifications are preliminary, be prepared for something a little better or maybe a little less. Anyway it will allow you to back off your hard disk to floppies and to also split large files across a disk. On recovery of backed up files it will allow them to be recombined as the original. This program proved very popular on the 6809 UniFLEX and FLEX systems. And has long been needed on OS-9.

1. ARCHIVE: Creates a special header to keep track of what is stored. Files may be merged for easier handling. Large files may be split according to user specifications.

2. RETRIEVE: This will recover files previously backed up. They will be reproduced as the original. All or part of the original backup will be allowed.

3. DIRARCH: This is a program to reveal where each is stored on the backup media. Also included will be pertinent information concerning the backed up files.

4. SPLIT: Merged files can be re-split with this program.

5. PARITY: Checks the integrity of backed up files.

If interested call or write S.E. MEDIA for availability and shipping.

DMW





## The Macintosh~Section

Reserved as a

**A place for your thoughts**

And ours.....

## **Mac-Watch**

Recently we received several new products I felt you might be interested in. Here are a couple for this month.

### **BATTERY PAK**

From: Batteries Included  
30 Mural Street, Richmond Hill  
Ontario L4B 1B5 Canada

For you Macintosh users, here is a nifty set of DAs (Desk Accessories) that can sure make things a lot simpler.

Battery Pak is an assortment of 9 DAs. As such they must be installed with the latest version of the DA/Font Mover. Once installed they are available by the Apple Pull Down Menu from any application.

While on the subject of DAs, let me give you a wee bit of warning. *Never, I say never, have two DAs active at the same time.* Most applications will not only groan, they will in fact, in most instances, die a most horrible death, right before your eyes. And what is even more important, before you get a chance to save whatever you happen to be working on. A most unpleasant situation! Now for more pleasant things.

First let me tell you that the documentation is very well done. A spiral bound 120 or so page manual lays it out so that even the new user will experience no trouble. Replete with a tutorial section as well as installation instructions (very precise).

Occasionally I hear from some new user who has a dread of attempting to install anything into the system file via the DA/Font Mover. Well, never fear, the instructions here are more than adequate for even the most timid.

### **PHONE PAD**

This is a hybrid telephone dialer, notepad accessory allowing an online phone book and dialer. Even from inside most application. Notes can be kept with names and telephone number. A total of 250 pages are supported. And search and find commands are available for getting a number or name quick.

### **PRINT TEXT**

This DA is a print spooler of sorts, that allows you to keep right on banging away at the keys as your ImageWriter is printing.

It is for text type files only (no MacWrite or Word full bore files). Text files may be printed with line numbers (very handy and worth the utility alone, only wished it worked on the LaserWriter), tab spaces can be reset, page headers added and 132 character per line output option selectable.

### **WINDOWS**

This sure can be a real help-mate. Windows is a real jewel for untangling those buried windows. A Windows Menu is printed and you can select the order of all or any windows open on the desktop.

### **CALENDAR**

This is a perpetual calendar that allows you to keep daily notes. A search feature allows rapid location of subject pattern anywhere in the calendar notepad. The dialer allows numbers in your notes to be dialed and you can even "quick" print selected entries to the ImageWriter.

The telephone dialer allows tone or pulse, special Hayes type commands are supported and control characters are there to create the proper amount of pause for those systems needing a pause between number or groups of number (MCI, etc.). tone or pulse may be switched within a series of numbers. How 'bout that?

The **CALENDAR MANAGER** has a wealth of functionality. It allows for note archives, clear notes, print notes (fonts, sizes, paper size, page-setup are as selected in the file). Other formatting functions make this DA far superior to the notepad that came with your Mac.

### **DISK TOOLS**

This DA prompts for the following functions: delete, rename, copy, drive info, eject and quit. All from within an application. There are several fail-safes built in to help avoid those mis-struck key strokes.

### **LAUNCHER**

The launcher lets you jump from one application to another without going back to the Finder each time, thus saving much valuable time.

Launcher has the following options, Finder, use DA, don't switch Systems, document, eject, launch, open folder and drive.

### **CALCULATORS**

Two calculators are provided. One a scientific and the other a RPN (Reverse Polish Notation) type, for you HP freaks. Both support the cut, copy, paste and clear functions of the Mac.

For the RPN calculator a handy and very complete (for this type application) instruction section, including some basic explanation of the RPN automatic memory stack. All about how simple RPN is (?). Anyway, for many it is the only "natural" way to figure expressions - so they say.

This is a very complete calculator and I remember not too many years ago I spent over \$300.00 hard earned

dollars to buy a calculator that was about par with this one. Functions include statistical - summation, mean, standard deviation and all the standard trig functions.

Both versions use the standard SANE environment, with 20 digits of precision. Each includes 10 memories and 4 stack levels plus last x.

So there you have it. A collection of DAs that should suffice for about

Oh, also a pretty important point - no copy protection. A real pain eliminated. And the price is right so there should be no need to "borrow" a copy from a buddy, the manual is worth the price alone.

Should you decide to purchase this package - tell them 68 Micro Journal sent you...and we thank you.

# MC68000/68010 SIMULATOR

## MC68000/68010 SIMULATOR

SIM68K is a nifty simulator that runs on the PC and executes 68000/68010 code. It is compatible with DOS 2.0 or higher. All of the 68000/10 instructions, addressing modes and condition codes are supported. It accepts files generated by a standard assembler (no assembler furnished).

Registers and memory can be examined and changed. Access to all registers, flags and memory is also supported.

To use this package you will also need, running on your PC, a text editor and an MC68000/10 assembler. Also, memory may be saved to disk.

SIM68K uses about 128Kbytes of RAM. So a minimum of at least 256K is needed. The entire 16 megabytes of addressing space of the 68000/10 is available for use. 52K of the PC's RAM is used to simulate 68000/10 memory. If more is needed it is broken into 4K blocks and written to a temporary disk file.

Hardware interrupts are not supported by SIM68K. However, the TRAP instruction may be used to cause the 68XXX program to execute as if it had received an interrupt.

Memory mapping of all 68XXX I/O simulation here is not practical. However, interactive I/O simplifies debugging of 68XXX programs, so SIM68K provides terminal I/O from within 68XXX programs.

## Command Syntax

SIM68K commands consists of two character command names followed by zero or more parameters. It is not case sensitive. All number are input in hexadecimal.

Commands supported are as follows: BA - Execute command, BK - breakpoint, DI - disassemble, DM - display memory, DR - display registers, EM - examine memory, EI - enable 68010 mode, EX - execute instructions, FM - fill memory with a constant, IO - disable special I/O, LB - load binary file, LO - load hex file, QU - exit SIM68K, Re - reset, SM - save memory to disk, SR - set register and SS - single step.

## BATCH command

Executes commands contained in a disk file. This command may be used with the PC command TYPE to display the commands that are to be executed by this command.

Four breakpoints are supported. The disassembler will disassemble all or parts of a file, according to the command line when disassemble is called.

Memory is displayed as hex and ASCII. This is in the normally accepted manner of RAM display routines. A line of hex code and then a column to the right that displays the printable ASCII character. Non-printable characters are printed as dots.

The display of registers is complete for either the 68000 or the 68010. In the case of the 68010 the additional registers - VBR, SFC and DFC, are shown.

Most of the other commands are self explanatory. However, it should be noted that the memory fill command fills a block of RAM with 16 bit data. And RAM will be filled starting at even address boundaries.

While this will not run as efficiently as if it were in an actual 68XXX system, it has its place for those who are using a PC type system and have a need to simulate the running of actual 68000/10 object code. It ran about 20 to 60 times slower than our 68008 system running at 10 Mhz. While this is really not a valid test of execution (simulation) speed on a PC, considering the inherent slow operation of the PC, it was not too far off our expectations.

Interested readers need to contact the vendor indicated below for additional information.

**BIG BANG Software, Inc.**  
POB 879  
Panama City Fl., 32402  
(904) 784-3393

A Staff Review

# XDMS

## Data Management System

a product of:  
Westchester Applied Business Systems  
2 Pea Pond Lane  
Briarcliff Manor, N.Y. 10510

After word processing, the most frequent serious use for a microprocessor is information storage and retrieval, since micros are especially well suited to these uses. Probably everybody who reads 68' Micro Journal has one or more editors or word processors, but perhaps no (or very limited) general purpose data management capabilities so I'm going to assume that some small discussion is in order.

We have come a long way since the first "Management Information Systems". My first exposure to one of these was many years ago when a representative of the comptrollers office proudly announced that I would be getting a "Management Information Report" every month. When the time came, I asked my secretary to pick it up. She returned shortly for the office boy. And later delivered the "report" to me — *all 3 cartons of 15 inch fanfold paper!* How I got them to stop that is a long boring story that would take too much space here. I just wanted to emphasize the point that "data" is not necessarily "information".

Information storage needs can range from very simple requirements that are best handled with a small card file (i.e. DON'T put it on computer!!) to enormously elaborate "data bases" that require large main-frame computer support and elaborate protocols to use them. An example of the latter would be the dial-it-with-a-modem subscription services (Compuserve, Delphi, etc.) or one of the more specialized services that many colleges now use to support students in their research efforts. The main advantage of computerized information storage is that this makes it possible to search for key words or phrases and cross references in minutes instead of the hours or days that would be required in conducting the search "by hand". Please note that somebody had to tediously punch all that information into the machine in the first place or a computerized search would not be possible at all. And this can be costly in time and effort. In many cases the data entry burden is great enough to overwhelm any potential retrieval convenience benefits and each application must be decided on its own merits. Hence my note above about the old fashioned card file — micros are frequently mentioned in connection with "recipe file", but my wife has never shown any interest in typing all her 100's of recipes into a computer.

XDMS is a general purpose data management system intended to satisfy requirements between the two extremes mentioned above or data storage needs up to the maximum capabilities of a micro. What does it do "right out of the box"? Nothing. XDMS is an interpreter in the same sense that BASIC is an interpreter. In fact, some of the more than

80 "key words" are the same as BASIC and experienced BASIC users should have little difficulty picking up the XDMS language. Learning XDMS is also made somewhat easier due to its well organized manual — each key word is thoroughly described on a separate page and presented in alphabetical order. There is also a convenient, memory jogger, keyword summary at the end of the manual.

After deciding on his initial application, the user must define the format and content of the data file using the system's integrated CREATE utility. While there is no tutorial on the design and use of data base systems, the manual includes step-by-step examples of checkbook and mailing list applications that are developed in a very orderly, instructive fashion. In fact, the presentation of the thought processes used in setting up the mailing list was one of the best I have seen anywhere. After following these examples through carefully, the new user should be well on his way toward designing his own, unique, applications.

That's what this is all about. After all, you can buy (or scrounge from public domain) checkbook and mailing list programs, right? But what about a data system that exactly fits some peculiar business requirements. After the initial effort in learning to use XDMS, it should be possible to satisfy most or all of a small business data management needs.

XDMS is an integrated system in that the various data management actions (such as CREATE, UPDATE) are all contained in a single .CMD file, a more convenient and compact arrangement than some systems that handle these separately. The exception is a separate configuration file that is loaded by XDMS each time it is used. This separate file MUST be on the system drive (DR0) or the XDMS will use its default parameters. The idea behind having a set of configuration parameters is one of the clever ideas included in the design of XDMS. Among other things, the XDMS.CFG file may be SET to define the dimensions of the terminal screen and the printer page so that output will be formatted correctly depending on which is has been directed to. I would advise reading the description of SET command before trying to follow the INSTALLATION instructions so as to understand some otherwise cryptic abbreviations and to notice that many additional options are available.

For example, SD, WD, PD, PD and DD stand for System Drive, Work Drive, Paging Drive and Data Drive; four possible drive assignments compared to the standard Flex System and Work drives. It should be noted here that a design objective for XDMS was to have it mask the characteristics of the operating system that supports it. That is, the system drive assignments and TTYSET parameters are

NOT used by XDMS and may, in fact, interfere with XDMS operation. Users are advised to turn off the system parameters so as to avoid conflicts. The use of ASN S=ALL especially seemed to upset XDMS. I personally felt this a bit inconvenient, but I suppose frequent users could develop a set of EXEC files to redefine and then restore the system parameters.

In keeping with the concept of masking the operating system character, no access to Flex utilities is provided. The intent here is to protect users from their own folly by preventing the possible use of a command that destroys lower memory, crashing the XDMS task in progress. Instead, XDMS has its own integrated general utilities -- LIST, COPY, DIR, EDIT, RENAME, and DELETE. These are somewhat primitive by current standards, but adequate for the intended functions.

Other integrated utilities perform internal functions of the system. CREATE, of course, is used to define characteristics of the data file in order to ready the system for an application. UPDATE, obviously, is the tool for inputting, reviewing and changing the data files. An extensive array of specialized commands are provided to make this operation as convenient as possible, including the optional use of a pair of "key-board macros" which can be defined as needed to enter a string of the user's choice -- a great help for entering the same data repeatedly. DESCRIBE is used to present a description of the database file, echoing back the design developed during the CREATE phase. Essential if you have a variety of data files. SHOW also presents the file format, but differs from DESCRIBE by being specifically oriented as a data input aid, giving the field labels and a dummy line of data. This should allow turning over the system to an inexperienced person to input data with a very minimum of explanation. PURGE will clear out the data in a data file (watch out!) but leave the format definitions intact, thus allowing for re-use of the file for new data, a convenience for separate files that can use the same design over and over as, for example, a month-by-month record system where it's desired to retire each prior month when no longer needed.

In keeping with the idea of assigning disc drives within the program, the Data Drive being separate, can be devoted in its entirety for the sake of handling very large data files. Accordingly, XDMS does not automatically preserve a .BAK file (that would cut the storage space in half!) so the user must assume the responsibility for maintaining back-up copies. XDMS automatically compresses data so that, for example, a 128 byte field will only occupy the space needed by the data it contains (plus a little for overhead), thus providing for even greater capacity. This process is transparent to the user and is, I think, a powerful argument for the technique. For those who insist on the "random access format" there is an option to write files to disk with their full data field lengths. In this mode the files are still processed serially by XDMS; an update results in the file being re-written, so the usual random write-directly-to-disk is not available.

Up to this point, we've been describing work, but now the fun part can begin. XDMS uses "process" files (.PRC default) to make use of the data stored in the data files and the

basic-like key words are actually application programs developed by the user to achieve the desired results. Space won't allow a description of each, but this brief summary of keywords may give some idea of the scope of the process:

```
INPUT, INCLUDE, READ, IF/AND/OR
-- Input file and record selection
LIST, SELECT, HIDE, STRING, ARRAY, MOVE
-- Input field selection
DEFINE, FILL, MERGE, JOIN
-- Auxiliary file/field selection
CAIC, SORT, SUM, BY, TOTAL
-- Output math, sorting and aggregation
ON, LINK, INSERT, SUBTOTAL, CLEAR, SKIP, EJECT
-- Subset processing
FORM, @ (line), > (file), < (field)
-- Form generation
TITLE, SPACE, DOUBLE-SPACE
-- Report formatting
NO-HEADING, NO-GROUPS, NO-x (labels)
-- Omit headings
DUPLICATE, FOLD, IND, SEP, TAB
-- Data formatting
COUNT, OUTPUT, RANDOM, SAVE, WRITE, PRINT, DISPLAY
-- Output options
CLRS, @ (X, Y), INKEY, FORK, EXEC
-- Terminal oriented process options
&& (symbolic), * (note), / (display)
-- Procedure file options
END, RUN, QUIT, NEW
-- Process execution/termination
```

Up to three data files may be used to form a "database" on which to act. The data is extracted selectively, may be merged with other data, may enter into calculations, SUMmed, written to output, etc. Additionally, special form files may be developed separately, customized to the application, (e.g. a printed business form, index cards, letters, etc. may form the basis for the design), and these called in as required to give the desired report.

In processing the data, a high degree of flexibility is provided in determining what is used as input, how the data is used in the report and, indeed, what is shown as the output. It is possible to reduce a large data file to a single page, or even a single line for that matter, by simply using the instructions that "hide" the data that is in use. Surely a busy boss would appreciate this kind of brevity, and if he feels he needs more detail, it's a relatively simple matter to provide that too.

*I expected XDMS to be fast -- it's written in assembler language -- and it is!* XDMS impressed me as being well done by a competent programmer and should be a valuable tool for the management of a micro-computer based business.

Art Weller

EOF

FOR THOSE WHO NEED TO KNOW

68 MICRO  
JOURNAL™

# Bit-Bucket



By: All of us

"Contribute Nothing - Expect Nothing", DMW '86

## loc

LOC (utility in UCS) is similar to FLEX's FIND command, except that the target is a sequence of (up to 16) bytes; the disk address (track/sector; T/S and index; BYTE #) of occurrence(s) is displayed. This is useful if one wants to replace (using EXAMINE) a few bytes in a .CMD or .BIN file, without having to modify the structure of the whole disk.

The assembler program in the accompanying listing was written for 6809 FLEX. In what follows, the documentation is expanded to cover a few crucial points which are essential to the operation of the program.

1) Comparison is carried over most simply in a stepwise fashion between the target sequence (in SEQ) and a sequence of equal length (value in CNT) copied from the disk file. In order to simplify the copy and compare process, a circular buffer of 256 bytes is implemented; it is accessed by X indexing with signed offset (in B register). To start a comparison (to begin with or after every match), a sequence of appropriate length is read from the disk file and copied to the middle (SEQ) of the buffer up. If and as long as the comparison fails, bytes are added at the end of the sequence. Note that wrapping around occurs automatically because of the signed offset: +127 is followed by -128.

2) In case of match, the disk address of the beginning of the sequence is printed (cf. AFFI). A complication arises when such a sequence overlaps distinct sectors. The solution is to keep track of the sectors succession: while reading every new byte (cf. FMSTS) for transferring to the buffer, whenever T/S changes ("current position" in FCB), the previous value is kept in TSPR. When a match is detected, the byte index of the start of the sequence is calculated: if it is larger than 4 it must be located in the current sector (leading 4

bytes in a sector are reserved for FLEX's use); otherwise, it means that the sequence origin is in the previous sector (cf. TSPR), with now the actual byte index = (current data index) - 1 - (CNT) + 252.

3) A minimum of processing must be done by the LIFI subroutine while reading a binary file. In particular, significant code sections are detected when the special header byte "\$02" is encountered, followed by 3 bytes (starting memory address and record length). The record length is memorized in NBLOC; it is updated when reading consecutive bytes in a record.

Jean LAVOREL  
"Santoline" Qu. Bigourdin D13  
13770 VENELLES France

list loc

```
* PROGRAM "LOC"
* - FINDS A SEQUENCE OF UP TO 16 BYTES
* IN A BINARY DISK FILE;
* - PRINTS THE DISK ADDRESS OF THE
* BEGINNING OF SEQUENCE.
* SYNTAX:
*     LOC <FILE> NN MM OO ... <CR>
*     NN,MM,OO ... : BYTES IN HEXADECIMAL
```

CHAR.

```
*
*     J. LAVOREL
*     MAY 1986
*
```

\* FLEX REFERENCES AND SUBROUTINES



```

GETFIL EQU $CD2D
SETEXT EQU $CD33
FMS EQU $D406
GETHEX EQU $CD42
PSTRNG EQU $CD1E
OUTADR EQU $CD45
PUTCBL EQU $CD18
OUTBEX EQU $CD3C
RPTERR EQU $CD3F
FMSCLS EQU $D403
WARMS EQU $CD03
FCB EQU $C840
LSTTRM EQU $CC11
EOL EQU $CC02

```

\* BEGINS IN UCS...

```

ORG $C100
LOC BRA LOC0
FCB 1 VERSION NO
CNT RMB 1 LENGTH OF SEQUENCE
SEQ RMB 16 SEQUENCE BUFFER
NBLOC RMB 1 LENGTH OF A BINARY
RECORD
TSPR RMB 2 PREVIOUS TRACK/SECT.

```

\* PROCESSES THE COMMAND LINE:  
 \* - GETS THE FILE'S SPECIFICATION  
 \* (EXT. DEFAULTS TO BIN.)  
 \* AND OPENS IT FOR READ;  
 \* - GETS SEQUENCE OF BYTES AND  
 \* STORES IT AT "SEQ";  
 \* - EXITS WITH ERROR IF SEQUENCE IS EMPTY.

```

LOC0 LDX #FCB
      JSR GETFIL FILE'S NAME?
      LBCS ERREUR
      CLRA
      JSR SETEXT DEFAULTS TO .BIN
      LDA #1
      STA 0,X
      JSR FMS OPEN FILE FOR READ
      LDA #FFF
      STA 59,X COMPRESSION FLAG OFF
      CLR CNT
      LDU #SEQ U POINTS TO SEQ.

```

BUFFER

```

OCT2 LDA LSTTRM
      CMPA #$0D
      BEQ OCT1
      CMPA EOL
      BEQ OCT1
      JSR GETHEX GET 1 BYTE FROM LINE
      BCS OCT1
      TFR X,D INPUT BYTE NOW IN B

```

REG.

```

STB 0,U+
INC CNT
BRA OCT2
OCT1 TST CNT
      BEQ ERREUR IF SEQUENCE EMPTY
      CLR NBLOC

```

\* ON ENTRY OR ANYTIME A SUCCESSFUL  
 \* COMPARISON WAS ACHIEVED,  
 \* A SEQUENCE OF (CNT) BYTES IS COPIED  
 \* FROM THE DISK FILE INTO  
 \* A CIRCULAR BUFFER.

```

AMORC LDX #TMP X POINTS MIDDLE OF
CIRCULAR BUFFER
      CLRB (COUNT OF BYTES KEPT
IN B)
      AMORC1 BSR LIFI READ 1 BYTE FROM FILE
      STA B,X AND TRANSFER TO CIRC.
BUFFER
      INCB
      CMPB CNT
      BNE AMORC1 UNTIL LENGTH =
TARGET'S
      BRA COMP THEN PROCEED WITH
COMPARISON.

```

\* COMPARISON MODULE  
 \* (CIRCULAR BUFFER IMPLEMENTED AS X  
 \* INDEXING WITH SIGNED OFFSET IN B):  
 \* - MAIN ENTRY AT "COMP" TO COMPARE  
 \* CURRENT SEQ. IN CIRC. BUFFER WITH  
 \* TARGET SEQ.;  
 \* - IF COMPARISON FAILS, UPDATES SEQ.  
 \* IN CIRC. BUFFER AT "COMP1",  
 \* - ELSE PRINTS DISK ADDRESS AND  
 \* CONTINUES.

```

COMP1 PULS B RESTORE B
      BSR LIFI (SEE "AMORC1" ABOVE)
      INCB
      STA B,X
      COMP PSHS B MAIN ENTRY POINT, SAVE
B

```

```

      SUBB CNT
      LEAY B,X Y POINTS CURRENT
SEQUENCE

```

```

      LDB CNT
      LDU #SEQ
      COMP2 LDA 0,U+ COMPARE
      CMPA 0,Y+
      BNE COMP1 IF FAILURE
      DECB
      BNE COMP2 TRY NEXT BYTE

```

```

PULS B      IF MATCH,
LBSR AFFI   PRINT DISK ADDRESS
BRA AMORC

* ERROR HANDLING AND EXIT
ERREUR LDX #MSER
        JSR PSTRNG
        BRA ERDI1
ERDI  LDA 1,X
        CMPA #8      EOF?
        BEQ ERDI1
        JSR RPTERR
ERDI1 JSR FMSCLS
        JMP WARMS
MSER  FCC 'SYNTAX ERROR',4

* SUBROUTINE TO READ THE BINARY DISK
* FILE 1 BYTE AT A TIME:
* - KEEPS READING AT EACH CALL UNTIL
* THE CURRENT RECORD IS EXHAUSTED;
* - LOOKS FOR NEXT RECDRD ($02 HEADER)
* AND STORES ITS LENGTH IN "NBLOC";
* - SKIPS TRANSFER ADDRESSES ($16 HEADER).
LIFI  PSHS X
        LDX #FCB
LIFI4  TST NBLOC
        BEQ LIFI1     CURRENT REC. EXHAUSTED
        BSR FMSTS     GO READ 1 BYTE
        BNE LIFI2     IF DISK READ ERROR
        DEC NBLOC
        PULS X,PC
LIFI2  LEAS 4,S      TO SKIP RETURN ADDRESS
        BRA ERDI
LIFI1  BSR FMSTS     LOOK FOR HEADERS
        BNE LIFI2
        CMPA #$16    "TRANSFER ADDRESS"?
        BNE LIFI3    NO
        BSR FMSTS     YES: SKIP 2 BYTES
        BNE LIFI2     (TR. ADDRESS)
        BSR FMSTS
        BNE LIFI2
        BRA LIFI1
LIFI3  CMPA #$02    "RECORD"?
        BNE LIFI1    NO
        BSR FMSTS     YES: SKIP STARTING
ADDRESS
        BNE LIFI2
        BSR FMSTS
        BNE LIFI2
        BSR FMSTS     READ RECORD LENGTH...
        BNE LIFI2
        STA NBLOC     ...AND KEEP IT.
        BRA LIFI4

```

```

* SUBROUTINE READS 1 BYTE THROUGH FMS,
* CHECKS WETHER READING GOES ACROSS
* A SECTOR BOUNDARY - IF SO, SAVES
* "CURRENT POSITION" INTO "TSPPR".
FMSTS  LDY 30,X      Y= "CURRENT POSITION"
        PSHS Y      SAVE IT ON STACK
        JSR FMS     READ 1 BYTE
        BEQ FMSTS1
        PULS Y,PC   IF ERROR, RETURN W.
"NE" CONDITION
FMSTS1 LDY 30,X      HAS "CURRENT
POSITION"...
        CMPY 0,S    ... CHANGED?
        BNE FMSTS2
        PULS Y,PC   NO: RETURN W. "EQU"
COND.
FMSTS2 PULS Y      YES
        STY TSPPR   SAVE FORMER "CUR.
POS."
        ORCC #04    Z=1
        RTS        ... WITH "EQU" COND.

* PRINTS THE DISK ADDRESS
* IN CASE OF MATCH.
AFFI  LDX #MSG
        JSR PSTRNG
        LDX #FCB
        LDA 34,X    A= "DATA INDEX"
        DECA
        SUBA CNT     NOW, A= START OF SEQ.
        CMPA #4      SAME SECTOR AS
CURRENT?
        BHI AFFI1    YES: GO PRINT...
        ADDA #252    NO: CORRECT FOR
PREVIOUS...
        PSHS A      ... SECTOR AND SAVE
IT.
        LDX #TSPPR   X POINTS PREVIOUS T/S
AFFI2  JSR OUTADR
        LDA #'        (SPACE)
        JSR PUTCHR
        TFR S,X      X POINTS SAVED DATA
IND.
        JSR OUTHEX
        LEAS 1,S     CLEAN STACK
        RTS
AFFI1  PSHS A
        LEAX 30,X    X POINTS CURRENT T/S
        BRA AFFI2
MSG  FCC 'TRACK/SECTOR, #BYTE ',4

* CIRCULAR BUFFER
* WITH MIDDLE LABELED BY "TMP"
RMB 128
TMP  RMB 128

* (TRANSFER ADDRESS)
END LOC

```



In U.S.A. Contact:  
Cosma Pabouctside  
GESPAC Inc.  
50 W. Hoover Ave  
Mesa, AZ 85202  
(602) 962-5559

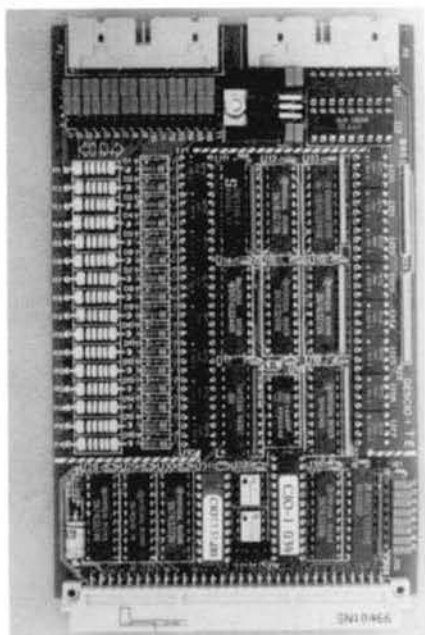
In Europe Contact:  
Bob White  
GESPAC SA  
3 Chemin des Aulx  
CH-1228 Geneva  
(022) 71 34 00

#### GESPAC INTRODUCES AN OPTOISOLATED PARALLEL INPUT/OUTPUT MODULE FOR THE G-64 BUS

MESA, AZ, February 25, 1987-GESPAC introduces an industrial parallel Input/Output module for the G-64 bus. The GESPIO-1 is built on a standard single height Eurocard and provides 16 Inputs and 16 Outputs.

All lines of the GESPIO-1 are totally isolated from the G-64 bus by optocouplers providing protection up to 1500 Volts. The small form factor of the single height Eurocard and its rugged DIN pin-in-socket connector make the GESPIO-1 particularly resistant to shock and vibrations. The board is ideal for a variety of machine and process control applications.

The output section of the GESPIO-1 is capable of driving 100 mA at 12 to 24 Volt. The input section is jumper configurable to accept active low or active high signal levels on each line. Six of the input lines are filtered through a debounce circuitry so that they can be used to generate interrupts directly on the G-64 bus.



The GESPIO-1 will work with any G-64 or G-96 bus compatible processor module. The G-64 bus is an easy-to-interface 16-bit bus aimed at midrange industrial applications.

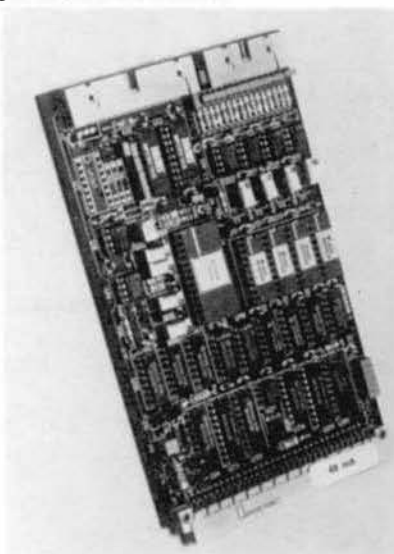
The GESPIO-1 is available from stock at the low unit price of \$395.

#### GESPAC INTRODUCES A 10-BIT A/D AND D/A CONVERTER MODULE FOR THE G-64 BUS

MESA, AZ, February 25, 1987-GESPAC introduces a complete Analog to Digital and Digital to Analog converter module for the G-64 bus. The GESADA-1A is built on a standard single height Eurocard and provides a 10-bit resolution.

The GESADA-1A uses relaxed 12-bit converter devices and thus provides a better than 0.1% accuracy. This level of precision is ideal for a variety of industrial applications in process and machine control. The small form factor of the G-64 card and its rugged DIN pin-in-socket connector make the GESADA-1A particularly resistant to shock and vibrations.

The Analog to Digital converter section of the GESADA-1A supports 16 single ended or 8 differential input channels. The Analog input range of the board is jumper selectable to  $\pm 5V$  or  $\pm 10V$  bipolar, or 0-10V unipolar. The converter uses a high precision sample and hold device and an optional instrumentation amplifier, capable of boosting the gain between 10 to 1000. The GESADC-1A will perform an Analog to Digital conversion in 35 microseconds.



The Digital to Analog section of the GESADA-1A provides 4 output channels, each with 10-bit  $\pm 0.25$  bit resolution. The output range is jumper selectable to  $\pm 10V$  bipolar or 0-10V unipolar. A conversion is performed in less than 4 microseconds and the maximum output current is 5mA.

The GESADA-1A will work with any G-64 or G-96 bus compatible processor module. The G-64 bus is an easy-to-interface 16-bit bus aimed at midrange industrial applications.

The GESADA-1A is available from stock at the low unit price of \$750 and \$795 for the version without and with instrumentation Amplifier respectively.

# CERTIFIED SOFTWARE CORPORATION

636 CAMINO CABALLO, NIPOMO, CA 93444 USA TELEPHONE 805-929-1395 TELEX 467013

## INTEGRATED PASCAL PACKAGE FOR 68000 SERIES PROCESSORS

Nipomo, CA., February 12, 1987 - Certified Software announces a major enhancement to its OmegaSoft 68000 Pascal Package first introduced in 1985.

The new release features a menu driven "Pascal Shell" that allows organization of project files and control of the other programs in the package. The shell allows access to the screen editor, compiler, assembler, linker, host debugger, and target debugger. All source files are available in a menu for the editor. Only pascal source files are shown in the compile menu, likewise, only assembly source files are shown in the assemble menu.

A unique feature of the supplied screen editor is the ability to compile or assemble directly from the editor's internal buffer, greatly enhancing the speed of syntax checking. The compiler and assembler also report back errors to the editor, resulting in automatic placement of the cursor for error correction.

The pascal language supported is based on the ISO standard and has extensions for real time applications. A few of the major extensions include additional data types and operators, modular compilation, exception and task procedures, and selective error trapping.

This package is available to run on OS-9/68000, PDOS, and VERASDOS based host computers. It is available directly from Certified Software, or through one of its European distributors. Priced from \$900.

### Continued From Last Month HEIR UNIX Modifications by Bradford Taylor

```
while(strcmp(fp->extension,"DIR")==0) {
    strcpy(fp->filename,fnam,0);
    return(to_int(4*(fp->start_track)));
}

/*
*** Make fns call and parse error
*/
do_fns()
{
    char error;

    error = _fns(fp->f); /* call file system */
    if(!fp->error)
        return(error);
    fns_err(fp); /* report error */
    exit(0); /* then die */
}

/*
*** Return track/sector of current directory
*/
current_dir()
{
    unsigned ts;

    fp->drive = drive;
    fp->function = IO_OPEN; /* Open DIR */
    do_fns();

    get_info();
    ts = to_int(4*(fp->buffer[24]));
    free_size = fp->size; /* remainder free size */

    return((ts*75+0x200));
}

/*
*** show directory until eof
*/
show_dir()
{
    char count=3;

    while(get_info() != -1 && fp->filename[0])
    {
        if(fp->filename[0] == DELETED) /* Not deleted */
        {
            if(mask && !match(mask,fp->filename))
                continue;
            if(mask && !match(mask,fp->extension))
                continue;
            if(count--<1) {
                space(2);
                size;
                fns_err();
                print(fp->filename,8);
                fns_err();
                print(fp->extension,3);
                fns_err(); /* display size */
            }
        }
    }
}
```

```

    fns_err();
    fns_err();
}

/*
*** print to either null char or to count
*/
print(s,n)
char *s;
int n;
{
    while(s && n-->0)
        fns_err(s);

    if(n > 0)
        space(n); /* pad rest of field */
}

/*
*** output a space
*/
space(n)
int n;
{
    while(n-->0)
        space();
}

space()
{
    fns_err(' ');
}

/*
*** match against key
*/
match(key,s)
char *key,*s;
{
    while(*key && *s == *key)
        ++key;
    return(!*key);
}

/*
*** Return track/sector of parent directory
*/
parent(ts)
unsigned ts;
{
    read_sector(ts); /* read parent directory */

    ts = to_int(4*(fp->buffer[4]));

    return((ts*75+0x200)); /* don't backup beyond home */

/*
*** convert 2 referenced bytes to an integer
*/
to_int(ip)
int *ip;
{
    return(*ip);
}

/*
*** Store integer into a two byte buffer
*/
store_int(ip,ts)
int *ip,ts;
{
    *ip = ts;
}

/*
*** do GET_INF file manager Command
*/
get_info()
{
    fp->function = GET_INF;
    return(do_fns());
}

/*
*** Read a sector
*/
read_sector(ts)
unsigned ts;
{
    fp->drive = drive; /* set drive number */
    fp->function = RD_SECT; /* set function */
    store_int(fp->current_track,ts);
    return(do_fns());
}
```

XOR

To Be Continued Next Month



Worstead Laboratories (Reg. Office)  
North Walsham, Norfolk NR28 9SA  
Tel: 106921 404086  
Telex: 975548 WMICRO G  
*MAPCON approved consultants*

### 32-BIT MC68020 BASED SINGLE BOARD COMPUTER FOR O.E.M.'s

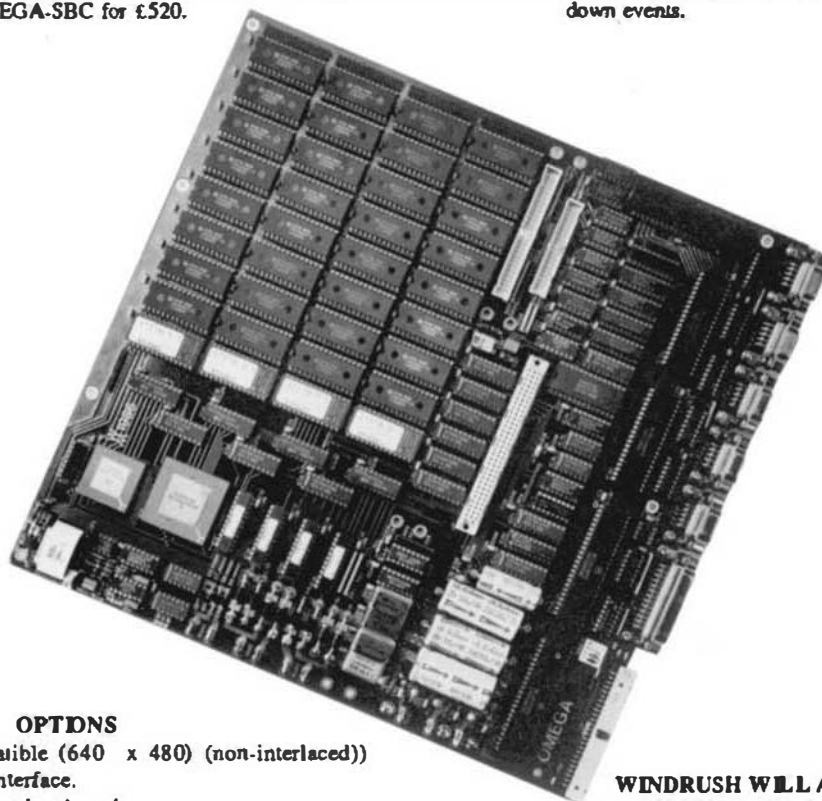
Windrush Micro Systems Ltd. are pleased to announce the immediate availability of their 'OMEGA-OEM' MC68020/MC68881 Single Board Computer to O.E.M.'s.

The multilayer PCB, which measures 12" x 12", and has been designed to be integrated into a 1U high 19" rack or custom designed OEM hardware. The cost of the OMEGA-SBC is £1895 1 off, £1795 2-5, £1695 6-9, £1595 10-24, £1495 25-99 and £1395 100+. The price falls below £1000 in 1000+ quantities.

To assist the OEM develop code for target boards a fully configured version of the OS-9/68K (tm Microware Systems Corporation) operating system complete with C compiler is available for the OMEGA-SBC for £520.

### PRODUCT FEATURES

- 12.5 MHZ MC68020 32-bit processor & MC68881 FPCP (speeds to 25 MHz optional)
- 1 megabyte of no wait-state, non-volatile 100% STATIC RAM. Expansion to 5 Mb optional. 128K/256K ROM capacity.
- Five RS-232c 'DCE' serial ports, centronics parallel printer port and a 16-bit bi-directional parallel printer port.
- Battery backed clocked calendar.
- SCSI initiator and a Shugart compatible floppy disk controller.
- Fully buffered 16-bit bus expansion provides 16 Megabytes for user expansion.
- 8 watt total power requirement. On-board rectifiers, filters and regulators allow all power requirements to be met by a small off-board transformer. On board power supply supervisor sequences power-up and power down events.



### OPTIONS

- + IBM PGA compatible (640 x 480) (non-interlaced)) colour graphics interface.
- + 9-port serial expansion board.
- + G-64...STE...IBM PC bus interface adaptor.
- + Prototyping board.

**WINDRUSH WILL ALSO CUSTOMIZE THE  
OMEGA-OEM TO SUIT INDIVIDUAL  
REQUIREMENTS.**



Don Williams  
68 MJ and South East Media  
5900 Cassandra Smith Road  
Hixson, Tennessee 37343

You may recall the recent problems I had with disk read errors on some software I purchased. I have now found some bugs in BAS-EDIT which caused me considerable trouble. Fortunately, as you provided the source code I was able to overcome some of the problems.

Background-Host system UniBoard 6809 with 2 8" disk, operating with the onboard console and NOT using a terminal at all. Flex system.

#### Problem 1 :-

The AFFIX command has an intermittent habit of hanging up when trying to do its first test to see if there is an ACIA at \$E004. This address is within the memory mapped screen area on the UniBoard. Whether or not the system hangs seems dependent on what utility last accessed the disk containing AFFIX. But even this is erratic. I have not fully resolved this other than to read, do dumps and discoveries of that drive and try again. Eventually you might get through this test. Suggested change - allow menu bypass of this test for users who KNOW they do not have an ACIA at \$E004.

Problem 2 :- After stating the indirect address of the user input no echo routine, you get a FLEX error #21 (illegal file name). The reason for this is that subroutine RAWIN corrupts the stack and the RTS at its end can send you about anywhere, in my case apparently to the disk drivers. This happens as there are 2 PUSHES B statement bracketing the access to the users routine. The latter of this should of course be a plus B. That fixes the problem.

Additional note, having lost much hair resolving these problems, I now find that I cannot install BAS-EDIT as the console routines on the UniBoard do not provide cursor - left, cursor - right, or cursor - up facilities. Pity that BAS-EDIT doesn't use the cursor movement command instead, though I presume this would be slower. I will rewrite the UniBoard monitor routines one day - It has some other bugs which need fixing anyway, but that job will have to wait. It's been waiting 3 years now!

Regards  
Kingsley Burlinson

cc John Roberts, Research Inst. Co.

*Editor's Note: Thanks for the fixes above. I guess the UniBoard has some differences. I know that there are a lot of them out there. Running both OS-9 and FLEX.*

*BAS-EDIT has not experienced many complaints. And if everyone who finds a bug, in any software, and works up a fix, thousands of us would certainly like to know!*

Thanks Kingsley, keep in touch.

DMW

Mr. Don Williams  
68 Micro Journal,  
5900 Cassandra Smith Rd,  
Hixson, TN 37343  
U.S.A.

164 Kent Albert Rd.  
Cantbury 3126  
Australia.  
February 1rd 1987

Dear Don,

I have been annoyed with the TSC Text Processor for some time now with its limitation of only being able to input 128 characters in the command line.

Example of my annoyance, is when using the following command line.

PR,1,ATFILE1.TXT,1,ATFILE2.TXT,1,ATFILE3.TXT,-----,1,MYFILEM.TXT

The number of files that can be processed by the TSC Processor is limited by the characters in the filenames, as the total length of the command line cannot exceed the 128 characters allowed by Flex.

I suppose that one could arrange for the file line buffer to be relocated to some more convenient, spare memory space, to give an increased line buffer length.

BUT there will still be a limit on the number of files that the text processor can call.

I have included with this letter, a program which I wrote to overcome this limitation.

The program is called 'PRCONBUF', and it makes the Text Processor think that it has a continuous buffer.

The program inputs filenames from a text file and passes them to the Text Processor.

Hence there is no realistic limit to the number of files that can be processed when using 'PRCONBUF' and the Text Processor.

This feature is excellent when writing a book, or a manual, as it allows the overall text to be broken up into several, smaller files, such as chapters.

Then using 'PRCONBUF' and a text command file, the Text Processor can process each file and any macros, etc., will still work throughout the processing job.

PRCONBUF can be preceded with the 'P' (PRINT), or 'O' (OUTPUT) FLEX commands.

The format for the text command file to be used with PRCONBUF is as follows:

1) PR 1 - Calls the Text Processor (ie 0,PR)

2) 1 - Name of the first file to be processed (ie 1,MYFILE1.TXT).

3) 2 - Name of the second file to be processed (ie 1,MYFILE2.TXT).

FURTHER LINES - more file names.

LAST LINE - \$ (FIRST character to be '\$' (without a space) is used to re-terminate processing.)

The text command file can be created using either the FLEX 'BUILD' or a TEXT EDITOR. Use of the text editor allows additions or corrections to be easily made to the text command file.

PRCONBUF was written to reside in a spare block of memory at \$F400 in my system, in use it modifies the FLEX GETFIL jump, to an address in PRCONBUF, the GETFIL jump address is restored at program termination or error reporting.

One of my future tasks will be to modify PRCONBUF or write a new program to allow simple JOB CONTROL, this appears to be an evolutionary step up from PRCONBUF.

I have included a sample text command file, and sample processor files with the source listing of PRCONBUF.

Many thanks for a good 68XX magazine.

Yours Faithfully,

Howard Wills

Howard Wills.

```
* TEXT PROCESSOR CONTINUOUS BUFFER
*
* DISK FILE NAME = PRCONBUF
*
* HOWARD WILLS 27/01/87
*
* VERSION 1.0 27/01/87
*
* THIS PROGRAM READS A DISK FILE AND PUTS
* THE INFORMATION INTO THE FLEX BUFFER. THE
* FLEX BUFFER POINTER IS UPDATED EACH TIME
* DATA IS PLACED INTO THE BUFFER.
* THE PROGRAM EXITS THROUGH THE RUAL FLEX
* GETFIL ROUTINE.
*
* FLEX9 SC020 CONTAINS 7E 00 16 (CHECK YOUR VERSION)
*
* ASSEMBLER DIRECTIVES
*
* OPT PAG
*
* BOWA SS LIST
*
CD03 WARMS EQU SC003 FLEX WARN START
CD06 RENTER EQU SC006 FLEX RENTER ROUTINE
CD20 GETFIL EQU SC020
D036 FLKGET EQU SD036 FLEX GETFIL ADDRESS (ACTUAL)
C040 STVFC0 EQU SC040
CD3F RPTERR EQU SC03F
CD33 SETEXT EQU SC033
D006 PMS EQU SD006
CC14 BUFFER EQU SC014
C080 BUFSZ EQU SC080 FLEX LINE BUFFER START
C0A0 BUFCOM EQU SC0A0 BUFFER COM START
C0FF BUFEWD EQU SC0FF FLEX LINE BUFFER END
*
F400 CONFCH EQU EEDF0E COMFCH+34 PCB NEXT CHARACTER POINTER
F401 CFMXT EQU COMFCH+35
F402 COMDAT EQU COMFCH+44 PCB DATA START
*
* START OF RAM
*
F400 RAMST EQU $F400 RT SYSTEM SPACE RAM
*
* ORG RAMST
*
```

```

P400 J0 08 START BRA START1 BRANCH AROUND
P402 01 KXREG PC0 03 VERSION NUMBER ETC
P403 KXREG RMB 2 KEEP X-REG STORE
P405 KXPTR RMB 2 KEEP BUFFER POINTER STORE
P407 XTEMP RMB 2 X-REG TEMPORARY STORE
P409 00 PASHUM PC0 000 PASS NUMBER FLAG
P40A BF P403 START1 STX ESEBC BEEP X-RBC
P40D B6 P406 LDA PASNUM CBECK PASS NUMBER
P410 26 33 BNE COMPAS BRANCH FOR COMBUF PASS
P412 DE C0A0 LDX $B0FC0B GET 4 RESP BUFFER POINTER

P415 BF P405 STX KXPTR FOR LATER USE

* OPEN FILE
P418 BE P4AF LDX $CONPCB POINT TO PCB
P41B BD CD2D JSR GETFIL & SET UP PCB
P41E 25 67 BCS ERROR
P420 B6 03 START2 LDA #1 SET UP FOR TEXT
P422 BD CD33 JBR BTEXT? EXTENSION
P425 B6 01 LDA $B01 OPEN FILE FOR
P427 07 P4AF STA COMPCB A READ
P42A BD D406 JSR PWB
P42D 26 30 BNE ERROR BRANCH IF ERROR
P42F BE P40 LDX (START1) DIVERT FLEX JUMP
P432 BF CD2E STX GETFIL+1 TO COMBUF PROGRAM
P435 BE P405 COMPAS STX KXPTR SET UP POINTERS
P438 BF CC14 STX BUFFER
P43B BF P407 STX XTEMP
P43E 0E P4AF COMP 1 LDX $CONPCB SET TO PCB
P441 BD D406 JSR PWB GET NEXT CHARACTER FROM FILE
P444 26 41 BNE ERROR
P446 B1 0D CMPA $B0D CHECK FOR CARRIAGE RETURN
P448 27 04 BBO PWB BRANCH IF C.R.
P44A BD 35 SR PUT CHARACTER INTO BUFFER
P44C 30 F0 BRA COMPAT BRANCH TO PUT MORE INTO BUFFER

* FOUND C.R.
P44E BE P4BF PWB LDA $CONDAT TEST IF NEXT CHARACTER
P451 F6 P401 CPMOXT IS THE TERMINATION
P454 3A LBR CHARACTER 'S'.
P455 E6 04 LDB X
P457 C1 24 CMPB $B07
P459 27 1D LDB $B07
P45B B6 2C LDB $B07 BRANCH IF END CHARACTER
P45D BD 42 JSR CHRBUF PUT A COMBUF INTO BUFFER
P45F BE P405 FIXUP LDX ESEPT FIX BUFFER POINTER
P462 BF CC14 STX KXPTR
P465 DE P403 LDX KXREG RESTORE X-REG
P468 B6 P409 LDA PASNUM TEST PASS NUMBER
P46B 27 03 BBO PERIT BRANCH FOR FLEX EXIT

* NORMAL EXIT
P46D 7B D036 EXIT JMP FLRGET RETURN VIA FLEX GETFIL
* FLEX EXIT
P470 7C P409 PERIT INC PASNUM CLEAR PASS NUMBER FLAG
P472 7E CD06 JMP BTEXT
* EOF FILE ROUTINE
P474 B6 BD XNOFIL DA $B0D PUT A C.B. INTO BUFFER
P478 BD 27 BSB CHRBUF

* NOW CLOSE COMBUF TEXT FILE
P47A BE P4AF LDX $CONPCB POINT TO COM UP PCB
P47D B6 04 LDA $B04 AND THEN
P47F 07 P4AF STA COMPCB
P482 BD D406 JSR PWB CLOSE FILE
P485 27 12 BBO RESFLX BRANCH IF NO ERROR
P487 BF P403 ERROR STX KXREG
P48A BE D036 LDX $FLRGET MORE FLEX GETFIL
P48D BF CD2E TX GETFIL+1
P490 B1 P403 LDX KXREG
P493 BD CD3F JSR AFTERM REPORT ERROR
P496 7E CD03 JMP WAMB

* RESTORE FLEX ADDRESSES
P499 B1 D036 RESFLX LDX $FLRGET RESTORE GETFIL
P49C BF CD2E STX GETFIL+1
P49F 20 B1 BRA FIXUP OO FIX UP POINTERS

* CHARACTER TO BUFFER ROUTINE
P4A1 BE P407 CHRBUF LDX XTEMP GET POINTER
P4A4 BC COFF CMPX $B0FEND TEST BUFFER END ?
P4A7 27 03 BBO CHRBUF BRANCH IF END
P4A9 A7 80 STA 0,X+ PUT CHARACTER INTO BUFFER
P4AB BF P407 STX XTEMP SAVE POINTER
P4AC 39 CHRBUF RTS

*
P4AF P4AF ZWOPRG EQU *
P4A7 ZWOPRG EQU 320 COMBUF PCB
P4B7 ZWOPRG EQU 1
P4B9 P4B9 COMBUF EQU 1
* END OF PROGRAM
*
END START2

```

0 ERROR(S) DETECTED

PR  
2.PRFILE1  
2.PRFILE2  
2.PRFILE3  
2.PRFILE4  
2.PRFILE5  
2.PRFILE6  
2.PRFILE7  
2.PRFILE8  
2.PRFILE9  
2.PRFILE10  
2.PRFILE11  
2.PRFILE12  
5

SYMBOL TABLE:

HOP OF C0A0	BUFEND COFF	SUPPTR CC14	BUFBY C0B0	CFDXT F4D1
CHRBUF F4A1	CHRBUF F4A2	CONBUF F4E7	CONBUF F4E7	CONBUF F4E7
COMPAT F400	COMPAS F435	ENDFIL F476	ENDPRG F4AF	ERRON F487
PRIT F470	FIXUP F45F	FLRGET D036	JWB D406	PWB F44E
GETFIL CD2D	KXPTR F405	KXREG F403	RESIT F46D	PASHUM F409
RAMST F400	RENTER CD06	RESFLX F499	RYTSDH CD3F	SETEXT C031
STABT F408	START1 F40A	START2 F420	STY PCB C040	WAMB C001
XTEMP F407				

20 Sunnyside Ave. Suite A217  
Mill Valley, California 94941

Dear Mr. Williams

The January 68 Micro Journal included a letter offering a better character set for the ZRT-80 terminal board. If you are interested I wrote a ZRT-80 handler under OS9 for use on the Color Computer. I did have to examine the OS9 supplied 80 column handler, as terminal I/O does not seem to follow the level 1 requirements. I wrote radio shack about this and they said it was O.K. to publish my ZRT-80 handler. Are you interested? I could try Rainbow, but the last program I submitted took months to come up with a rejection. By the way, I drive the ZRT-80 via one of the CoCo PIA lines at 19.2x baud. That and 6 millisecond 80 track floppy makes OS9 a pleasure to use.

If I read your submission requirements correctly, a disk using the CoCo disk operating system is not acceptable. That is a shame, as the program that Rainbow rejected draws an interesting figure called the GOSPER curve. Many years ago Martin Gardner published an article in Scientific American on so called "Monster curves" that had caused a stir in mathematics. Recently Bythe published a program (in basic) to draw one of them; the Hilbert Curve. The Gosper curve was also in the article (in Scientific American) but I have never seen code for it. I believe the curve is also shown in Felix Mandelbrots' book on Fractals. Just think a chance to scoop Bythe; and my program is even more recursive.

An observation on the CoCo-3. Close, but no cigar. Basic is definitely slower than the CoCo-2, and they do not seem to have a technical manual so that we machine language programmers can use the new Bells and Whistles.

I like the idea behind the HEIR program for multiple Flex directories, but \$70 dollars is steep. I started to do multiple directories once, but got a little confused by some of the features in the Frank Hogg version of Flex. My main worry was over the changing of disk, and of course the accidental deleting of a directory pointer. About then I was given some patches for OS9 that made my 80 track 6 millisecond disk usable, and the change was so dramatic that I abandoned the work on the Flex.

Yours;  
Frank J. Wilson

Editor's Note: Frank, thanks for the letter. There are many of our reader who are also CoCo users. I bet some of them will be very interested in your ZRT-80 handler. Send it on!

While we normally do not accept CoCo formats, this time, send it on and I will get it over into our format. Who can pass up a 'scoop'?

Maybe Tandy will wise up and fix the glitches. I hear a lot of moaning, but the CoColli is just getting to base one. I say my thing a few months back. Time will tell. Hope it makes it!

As to the HEIR disk program. It has been a very popular piece of software. Sorry about the price, it just went up another ten buck lately. Also, the Data-Comp version of CoCo FLEX seems to work fine.

DMW

## The P68000 uLAB™

The P68000 uLAB™ microprocessor development system with an educational price of \$197.50 has just been announced by University Research and Development Associates, Inc. (URDA, Inc.) and is being manufactured by Quasitronics, Inc. The P68000 uLAB™ is a 68000 microprocessor and support chips with keypad, LED display, 4 K bytes SRAM, 8 K bytes of EPROM, cassette interface, software operating system, power supply, User's Manual and Programmer's Reference Manual completely packaged in a 2 inch 3 ring binder, i.e., a Notebook Computer™.

Traditional development systems require many times more cash expenditure, laboratory space, laboratory monitoring and scheduled access. The uLAB™ overcomes all of these limitations at a price approaching that of a textbook. The uLAB™ is engineered to be low cost providing a complete microprocessor development system for hands on experience for engineers, technicians and students.

The User's Manual contains a complete description of the system operation including explanation of all key functions, memory map, software utilities, and programming examples including Sound (Tone) Generation, Visual Display, and Timing Clock examples. The Programmer's Reference Manual includes the complete set of object (machine) codes for all 68000 instructions.

The user can conveniently carry the uLAB™ to the home, dormitory, or other work or study location to experiment with the microprocessor at his/her convenience providing laboratory type capability or it can be used in conjunction with lecture type courses without the additional laboratory expense and inconvenience.

In the future, a multiplicity of different uLAB™ microprocessor development systems will provide for inexpensive simultaneous access to many processors that is only available on a "one at a time" basis from expensive multiple microprocessor development systems.

Future add-on and expansion units available for individual uLAB™ systems allow the engineer, student or experimenter to "hardwire" the microcomputer to actual physical devices, e.g., LEDs, switches, motors, actuators, etc., thus providing a real world aspect to an education process involving interfacing, signal processing and control. The cost risk to the uLAB™ is minimal in that the chips involved in potential wiring mistakes are socketed thus providing simple repair mechanisms.

The P68000 uLAB™ can also be used as a self paced or self teaching material. The entire software monitor listing is included in the instruction manual along with comments and documentation. A schematic including every component is also included.

### A Summary of the P68000 uLAB™ features:

- 68000 16/32 bit microprocessor with necessary support (clock, memory, software), with a 26 Key user keyboard
- 4 K bytes Monitor and Utility EPROM
- 4 K bytes User expansion EPROM
- 4 K bytes of static RAM
- An 8 digit, 7 segment hexadecimal display
- Cassette tape interface
- A calculator type power supply
- Detailed User's Manual including schematics, programming instructions, and operating system listing
- A convenient 3 ring binder that stores and carries all of the above materials

The P68000 uLAB™ lists for \$295.00 with discounts to \$197.50 for Educational Institutions, Faculty and Students. To order, contact

Quasitronics, Inc. 1-800-245-4192, in PA 412-745-2663

For technical information, contact URDA, Inc. 412-683-8732



**MOTOROLA INC.**

**Microprocessor Products Group  
6501 William Cannon Drive West  
Austin, Texas 78735-8598**

### EDITORIAL CONTACT:

Mark Verduyn  
512/928-6804

### READER CONTACT:

Laura Tolpen  
512/440-2035

### INQUIRY RESPONSE:

R.O. Green  
P.O. Box 52073  
Phoenix, AZ 85073

## MOTOROLA ADDS THE M68HC05EVM TO THEIR DEVELOPMENT SYSTEM PRODUCT LINE TO SUPPORT THE 8-BIT M68HC05 MCU

Austin, Texas, January 23, 1987... Motorola's Microcomputer Operations adds the M68HC05EVM to their low cost Development System Family. These fully assembled boards are available now for \$500.00. Place orders through your Local Motorola Sales Office or Distributor.

The M68HC05 Evaluation Module (EVM) provides a tool for designing, debugging, and evaluating the M68HC05C4/C8 MCUs and future M68HC05 Family derivatives. By providing all of the essential MCU timing and I/O circuitry, the EVM simplifies user evaluation of the prototype hardware/software product. Entering data, program debugging, and EEPROM MCU programming is accomplished by the monitor ROM firmware via an external RS-232C compatible terminal connected to the EVM terminal port connector. A fixed 9600 baud rate is provided for the terminal port, and a software selectable 180-19.2K baud rate selection is provided for the host port. The EVM can interface virtually any development system including IBM PCs, VAXs, and PDP11s, as well as any Motorola development system.

MCU code may be generated using the resident one-line assembler/disassembler, or may be downloaded to the user program RAM through the host or terminal port connectors. User code may then be executed using various debugging commands in the monitor. User code may also be executed using the user reset switch. 40-, 52-, and 68-pin MCU I/O port connectors facilitate interconnection of the EVM to the target system for evaluation purposes. MCU device ROM is simulated by write protecting program RAM during program execution.

Jumper selectable options such as IRQ sensitivity and clock input selection are provided on the EVM, as well as an EEPROM MCU programmer. The EEPROM MCU programmer, under monitor firmware control, enables the user to check, erase, program, verify, and copy the contents of either 40-pin Dual-in-line (DIP) or 44-/52-lead Plastic Leaded Chip Carrier (PLCC) package EEPROM MCU devices. Switches allow user control of the reset and abort functions, and EEPROM MCU programming functions. The EVM requires a user-supplied +5, +12, and -12 Vdc power supply and an RS-232C compatible terminal for operation.



FORCE COMPUTERS, INC.  
727 University Ave.  
Los Gatos, California 95030

Prepared By:  
Shohat & Kahn PR  
Marry Shohat (408) 438-3484

CONTACT  
Force USA: Wayne Fischer (408) 354-3418  
Force GmbH: Anton Nausch (089) 608-910

Force Computers GmbH  
Daimlerstrasse 9, D-8012  
Ottobrunn, West Germany

Intelligent Mass Memory Controller  
Implements SCSI and Floppy Interfaces

10MHz 68010 and Firmware Maintains High Performance  
in VMEbus Applications; No Wait State Penalties

LOS GATOS, CA., March 3, 1987 — An intelligent mass memory controller for SCSI applications in VMEbus environments was announced today by Force Computers. The ISCSI-1 is a 10MHz controller based on a 68010 microprocessor with high-speed DMA channels coupled to SCSI and floppy disk controller chips. The board has many design innovations that boost performance.

Although its primary use is the high-speed transfer of data to and from Winchester and floppy disk drives, the ISCSI-1 is equipped to handle any SCSI-based application. A 128Kbyte dual-ported RAM with an innovative arbitration mechanism affords continuous access by the board's DMA controller without wait states and independent from all VMEbus accesses. This enables use of the ISCSI-1 in the most demanding application at the maximum data transfer rate (1.5Mbytes/second).

Given the growing popularity of the SCSI standard, this new board is likely to see expanding use with Winchester and optical drives, floppy and tape drives, laser printers, plotters, scanners and other performance-based peripheral devices into which SCSI controllers are being embedded. The ISCSI-1 is capable of acting as a host controller to any number of SCSI devices and up to four floppy drives, simultaneously. This is an unprecedented level of support for a single-board product on the VMEbus.

#### Not Just A SCSI-On-VME Controller

The ISCSI-1 is one of the most performance-laden peripheral controllers ever introduced (see Data Sheet). In addition to a 10MHz 68010 CPU, the board uses a speed-matched four-channel 68450 direct memory access controller (DMAC). SCSI control is handled by NCR's programmable 5386S chip, and floppy control is based on Western Digital's WD 1772 for connection to up to four devices. A parallel interface/timer chip (68230) affords local control and watchdog services, and a bus interrupter module (68153) services four IRQ lines to handle the fully asynchronous VMEbus interface via the PI connector. Residing between the VMEbus and the DMAC is 128Kbytes of dual ported static RAM with special arbitration logic. Extensive firmware is contained in 128Kbytes of EPROM.

The board employs many programmable logic devices to reduce chip count while increasing performance. Wherever required, standard logic devices are the fastest available.

#### Firmware Is Part of Performance

All activities to/from SCSI and floppy devices are handled by EPROM-resident firmware. The firmware package executes commands which are placed in the dual-ported RAM, and returns control and error messages to the host processor. Firmware modules provide initialization of input/output, control of SCSI initiator and target modes, command chaining and block buffering and hashing.

The command repertoire fully supports the SCSI standard; command options can be installed for support of any number of logical units. The firmware provides emulation of COPY, COMPARE, SEARCH and BACKUP commands. The repertoire also includes RESERVE/RELEASE and DISCONNECT/RESELECT. Similar capabilities are provided for the floppy drive interface.

The ISCSI-1 firmware provides a double software interface (two command blocks and buffers) which supports multi-processor access. The interfacing sector size between a host and the ISCSI-1 can differ from the physical sector size of the logical units. Up to five logical units can be under local control.

Users wishing to customize the firmware for specific applications can order the source code and its documentation from Force.

#### Back Panel Option for Device Connection

An optional back panel that plugs into the VMEbus P2 connector enables fast connection of SCSI and floppy devices via industry-standard flat cable. The back panel is a connector board that ties the P2 connector's user I/O lines to a two-row SCSI connector and a 34-pin, two-row floppy connector.

#### Price and Availability

The ISCSI-1 is in full production and available for immediate shipment in quantity. It is priced at \$3,990 (1-9). Volume discounts are available. The optional Back Panel is also available, and is priced at \$195 (1-9). Source code and documentation for the ISCSI-1's firmware is priced at \$3950.

#### About Force Computers

The leading independent designer and manufacturer of VMEbus products, Force is now in its sixth year of operation. The company has completed 16 consecutive quarters of profitable operation. Force is headquartered in Los Gatos, California with subsidiaries in West Germany, France and the United Kingdom. Sales, service and product support are provided on a worldwide basis.

**Classifieds** As submitted - No Guarantees - As is!

#### DAISY WHEEL PRINTERS

Quine Sprint 9 - \$900 Quine Sprint 5 - \$800

Winchester 10 Megabyte Drive - Seagate Model #412 \$275.

3 - Dual 8" drive enclosure with power supply. New in box. \$125 each.

5 - Siemens 8" Disk Drives. \$100 each.

Tandem Outpost II, 56K, 2 5" DSDD Drives, FLEX, MUMPS \$495.

TELETYPE Model 43 PRINTER - with serial (RS232) interface and full ASCII keyboard. \$250 ready to run.

SWTPC S/09 with Motorola 128K RAM, 1-MPS2, 1-Parallel Port, MP-09 CU Card - \$900 complete.

CDS-1. 20 Meg Hard Disk System with controller \$500.

(615) 842-4600 M-F 9 AM to 5 PM EST

\*\*\*

2-750k GIMIX sys III /w 19mb drives and 9 intelligent serial ports, OS-9. \$3600 each (negotiable).

Don Anderson 800-233-9309 in PA 800-632-1910

\*\*\*

FHL QT, 68008, 512k, 20 mega drive, 2 parallel, 4 serial OS-9 /68K, Dynacalc, Stylo, Basic 09. \$2000.  
Evenings 302-995-2298

FOR THOSE WHO NEED TO KNOW

68 MICRO  
JOURNAL™

another URDA quality product  
**P68000  $\mu$ LAB™**  
**-Notebook Computer™**

**\$197.50**

**EDUCATIONAL AND STUDENT PRICE**



The Power of a **68000** in one compact and efficient unit  
for hardware interfacing and machine language programming

***(a self-contained laboratory)***

**FEATURES:**

- 68000  $\mu$ P
- 4 K SRAM
- 8 K EPROM
- LED Display
- Integral Keypad
- Operating System
- Cassette Interface

**OPTIONS:**

- PADC-DAC-8  $\mu$ LAB™  
8 Ch A/D, 8 Ch D/A, 8 Bits
- Wire Wrap Expansion Board
- Digital Signal Processors
- and more . . .

*$\mu$ LAB™ and Notebook Computer™  
are trademarks of URDA*

**TO ORDER: Call QUASITRONICS**  
**at 1-800-245-4192    IN PA (412)-745-2663**

QUASITRONICS

211 Vandale Dr.

Houston, PA 15342

# Call for Software: 68000, C, Basic09 Sculptor

We are receiving calls and letters from numerous sources, including users, business and others looking for OS-9 68000 software; applications, etc.

Many of you have developed software that with little change could be adapted for others. If you are interested in selling it, please let us know. There is a growing market out there now. Get in on the ground floor!

If you can use additional income and have something that might be of interest, call and talk to Larry or Don.

## S.E. MEDIA Division - CPI

POB 849  
Hixson, TN 37343

Telephone (615) 842-6809  
Telex (510) 600-6630

# SK★DOS™

The Generic DOS™ for 68000 applications in

- ★ Industrial Control
- ★ Business Use
- ★ Educational Computing
- ★ Scientific Computing
- ★ Number Crunching
- ★ Dedicated Systems
- ★ Turnkey Systems
- ★ Data Collection
- ★ Single-board Computers
- ★ Bus-oriented Computers
- ★ Graphics Workstations
- ★ One-of-a-kind Systems
- ★ Advanced Hobbyist Use

SK-DOS is a single-user disk operating system for computers using Motorola 32 bit CPUs such as the 68008, 68000, 68010, and 68020. It provides the power of a full DOS, yet is simple and easy to use, and will run on systems from 32K to 16 megabytes. Because SK-DOS is easily implemented on a new system, we call it "The Generic DOS" which allows programs written for one system to be run on many others.

SK-DOS comes with over 40 commands and system programs, including a 6809 emulator which allows 88K SK-DOS to run application programs and languages developed for 6809. SK-DOS and other systems, Assemblers, editors, and higher level language support are available from third party software vendors and through public domain software.

SK-DOS is available for single copy or dealer sales, as well as OEM licensing. Single copies cost \$125 (inquire as to available systems). Extremely attractive OEM licensing terms are also available. An optional Configuration Kit contains a detailed Configuration Manual and two disks of source code for system adaptation, including source code for a system monitor/debug ROM and other programs useful for adapting SK-DOS to new systems.



## SOFTWARE FOR 680x AND MSDOS

### SUPER SLEUTH DISASSEMBLERS

**EACH \$99-FLEX \$101-OS/9 \$100-UNIFLEX**  
OBJECT-ONLY versions: EACH \$50-FLEX, OS/9, COCO  
interactively generate source on disk with labels, include xref, binary editing  
specify 6800, 1, 2, 3, 5, 8, 9/6502 version or Z80/8080, 5 version  
OS/9 version also processes FLEX format object file under OS/9  
COCO DOS available in 6800, 1, 2, 3, 5, 8, 9/6502 version (not Z80/8080, 5) only  
NEW: 68010 disassembler \$100-FLEX, OS/9, UNIFLEX, OS/9-68K, MSDOS

### CROSS-ASSEMBLERS WITH MACRO CAPABILITIES

**EACH \$50-FLEX, OS/9, UNIFLEX, MSDOS ANY 3 \$100 ALL \$200**  
specify for 180x, 6502, 6801, 6804, 6805, 6809, Z8, Z80, 8048, 8051, 8085, 88000  
modular cross-assemblers in C, with load/unload utilities NOW: OS/9-68K  
8-bit (not 68000) sources for additional \$50 each, \$100 for 3, \$300 for all

### DEBUGGING SIMULATORS FOR POPULAR 8-BIT MICROPROCESSORS

**EACH \$75-FLEX \$100-OS/9 \$80-UNIFLEX**  
OBJECT-ONLY versions: EACH \$50-COCO FLEX, COCO OS/9  
interactively simulate processors, include disassembly/formating, binary editing  
specify for 6800/1, (14)6805, 6502, 6809 OS/9, Z80 FLEX

### ASSEMBLER CODE TRANSLATORS FOR 6502, 6800/1, 68

6502 to 6809 \$75-FLEX \$85-OS/9 \$80-UNIFLEX  
6800/1 to 6809 \$60-OS/9 \$60-UNIFLEX

### FULL-SCREEN X BASIC PROGRAMS with cursor control AVAILABLE FOR FLEX, UNIFLEX, AND MSDOS

DISPLAY GENERATOR/DOCUMENTOR	\$50 w/source, \$25 without
MAILING LIST SYSTEM	\$100 w/source, \$50 without
INVENTORY WITH MRP	\$100 w/source, \$50 without
TABULA RASA SPREADSHEET	\$100 w/source, \$50 without

### DISK AND X BASIC UTILITY PROGRAM LIBRARY

**\$50-FLEX \$30-UNIFLEX/MSDOS**  
edit disk sector, sort directory, maintain master catalog, do disk sorts,  
resequence some or all of BASIC program, xref BASIC program, etc.  
non-FLEX versions include sort and resequence only

### MODEM TELECOMMUNICATIONS PROGRAM

**\$100-FLEX, OS/9, UNIFLEX, MS-DOS, OS/9-68K, UNIX**  
OBJECT-ONLY versions: EACH \$50  
menu driven with terminal mode, file transfer, MODEM7, XON-XOFF, etc.  
for COCO and non-COCO, drives internal COCO modem port up to 2400 Baud

## DISKETTES & SERVICES

### 5.25" DISKETTES

**EACH 10 PACK \$12.50-SSSD/SSDD/DSDD**  
American-made, guaranteed 100% quality, with Tyvek jackets, hub rings, and labels

### ADDITIONAL SERVICES FOR THE COMPUTING COMMUNITY CUSTOMIZED PROGRAMMING

We will customize any of the programs described in this advertisement or in our brochure for specialized customer use or to cover new processors; the charge for such customization depends upon the marketability of the modifications.

### CONTRACT PROGRAMMING

We will create new programs or modify existing programs on a contract basis, a service we have provided for over twenty years; the computers on which we have performed contract programming include most popular models of mainframes, including IBM, Burroughs, Univac, Honeywell, most popular models of minicomputers, including DEC, IBM, DG, HP, AT&T, and most popular brands of microcomputers, including 6800/1, 6809, Z80, 6302, 68000, using most appropriate languages and operating systems, on systems ranging in size from large telecommunications to single board controllers; the charge for contract programming is usually by the hour or by the task.

### CONSULTING

We offer a wide range of business and technical consulting services, including seminars, advice, training, and design, on any topic related to computers; the charge for consulting is normally based upon time, travel, and expenses.

Computer Systems Consultants, Inc.  
1454 Latta Lane, Conyers, GA 302 7  
Telephone 404-483-4570 or 1717

We take orders at any time, but plan long discussions after 5, if possible.

Contact us about catalog, dealer, discounts, and services.  
Most programs in source: give computer, OS, disk size.  
25% off multiple purchases of same program on one order.  
VISA and MASTER CARD accepted; US funds only, please.  
Add GA sales tax (if in GA) and 5% shipping.  
(UNIFLEX in Technical Systems Computers; OS/9 Microvare, COCO Tandy/MSDOS Microsoft)



**Now!** "TOPS  
*The OFFICE PRINT Shop™*"

Makes professionals of us all.  
For less than just affordable!

# DeskTop Publishing

**NOTE:** The following includes a 3 day, *hands-on*, instructional session for the entire "The OFFICE PRINT Shop™" system. A full 6 months, *no extra charge*, telephone or in-house (our offices, normal business hours) follow-up advisory service. We feature full Apple™ service and also national service by Honeywell. This includes *next day*, on site service. Over 95% of all service requests are completed on the initial call. *We understand the importance of fast service!*

100

**System 100** *Very Affordable Save up to 90% on your printing*

## The TOPS System 100

consists of the following items:

- A special Apple Macintosh Plus™ 1 Megabyte computer, including a double sided, double density 800,000+ character disk drive.  
An Apple LaserWriter™ typeset quality printer.

*Page make-up software  
galley typesetting software  
3 day - on site - instructions  
6 months instructional support &  
much more!*



### Option:

We also offer software to drive most all the large commercial typesetters. You can save a bundle by doing your own typesetting and proofing, and then downloading to the commercial system.

Leasing with payout available for all systems.



Also available with 20 million character storage 'Hard Disk'

## Data-Comp Division

CP  
I



A Decade of Quality Service

Systems World-Wide

**Computer Publishing, Inc.** 5900 Cassandra Smith Road  
Telephone 615 842-4601 - Telex 510 600-6630 Hixson, Tn 37343

This document composed, typeset and printed with a TOPS System 100

TOPS The OFFICE PRINT Shop is a trademark of Computer Publishing, Inc.  
Apple Macintosh Plus is a trademark of Apple Computer Company, Inc.  
LaserWriter is a trademark of Apple Computer Company, Inc.



# OS-9 UniFLEX MUSTANG-020, 68020, 68881 AND MORE HANDS-ON EXPERIENCE NOW SK\*DOS available

The DATA-Comp Division of Computer Publishing Corporation announces their new and innovative HANDS-ON 68020 computer familiarization two day event. A chance to TRY BEFORE YOU BUY!

For two full days (Monday through Friday - excluding legal holidays) each participant will be furnished the exclusive use of a 68020 computer (MUSTANG-020). Each system will have available native C compilers, BASIC, assembler and other high level languages. Each system will be equipped with the Motorola MC 68881 math co-processor, where applicable.

Each demonstration room will contain not more than two work stations. Each system will be equipped with floppy disk, 20 megabyte winchester technology hard disk, and 2 megabyte of RAM. RAM is partitioned as 690K bytes of RAM disk and 1.2 megabyte of user RAM space.

Participants are encouraged to bring along any source level projects, for evaluation, in C, BASIC or assembler. Call for availability of other HHLs.

Although this is not a training seminar, Data-Comp personnel are available for assistance and consultation. This event is scheduled for hands-on evaluations of the 68020 CPU, 68881 math co-processor and MUSTANG-020 system, operating in a functional environment.

Transportation to and from the airport and hotel/motel will be provided. Lunch provided both days. Chattanooga airport is serviced by American, Delta, Republic and other airlines.



## COST

One person - \$375.00

Two persons - \$595.00

\* Motel single \$22.00, double \$26.00  
Includes satellite TV - convenient to food and shopping



## DATA-COMP

A Division of  
Computer Publishing, Inc.

5900 Cassandra Smith Road  
Hixson, Tn 37343  
Telephone 615 842-4600  
Telex 510 600-6630

Systems available for both OS-9 and UniFLEX. Reservation should be made 15 days in advance. Attendee should initially indicate OS-9, UniFLEX or both. Special facilities available on request. Please write or call for additional information.

NOTE: Both OS-9 and UniFLEX are Unix type operating systems. Each as been enhanced in some aspect or another. Prospective attendees should have some working knowledge or experience with one of these operating systems, to gain full benefit of the session. However, a newcomer will find that it is a simple matter to be fairly proficient in using these systems in the allocated time. Special system instruction available on request. Call or write.

\* Hotel/Motel cost are separate cost, not included in the basic cost shown.

# THE 6800-6809 BOOKS

..HEAR YE.....HEAR

## OS-9™ User Notes

By: Peter Dibble

The publishers of 68' Micro Journal are proud to make available the publication of Peter Dibble's **OS9 USER NOTES**

Information for the BEGINNER to the PRO,  
Regular or CoCo OS9

### Using OS9

HELP, HINTS, PROBLEMS, REVIEWS, SUGGESTIONS, COMPLAINTS,  
OS9 STANDARDS, Generating a New Bootstrap, Building a  
new System Disk, OS9 Users Group, etc.

### Program interfacing to OS9

DEVICE DESCRIPTORS, DIRECTORIES, "FORKS", PROTECTION,  
"SUSPEND STATE", "PIPES", "INPUT/OUTPUT SYSTEM", etc.

### Programming Languages

Assembly Language Programs and Interfacing; Basic09, C,  
Pascal, and Cobol reviews, programs, and uses; etc.

### Disks Include

No typing all the Source Listings in. Source Code and,  
where applicable, assembled or compiled Operating  
Programs. The Source and the Discussions in the  
Columns can be used "as is", or as a "Starting Point"  
for developing your OWN more powerful Programs.  
Programs sometimes use multiple Languages such as a  
short Assembly Language Routine for reading a  
Directory, which is then "piped" to a Basic09 Routine  
for output formatting, etc.

## BOOK \$9.95

Typeset -- w/ Source Listings  
(3-Hole Punched; 8 x 11)

Deluxe Binder - - - - - \$5.50

### All Source Listings on Disk

1-8" SS, SD Disk - - - - \$14.95

2-5" SS, DD Disk - - - - \$24.95

Shipping & Handling \$3.50 per Book, \$2.50 per Disk set

Foreign Orders Add \$4.50 Surface Mail  
or \$7.00 Air Mail

If paying by check - Please allow 4-6 weeks delivery

\* All Currency in U.S. Dollars

Continually Updated In 68 Micro Journal Monthly

Computer Publishing Inc.  
5900 Cassandra Smith Rd.  
Hixson, TN 37343



\*FLEX is a trademark of Technical Systems Consultants

\*OS9 is a trademark of Microware and Motorola

\*68' Micro Journal is a trademark of Computer Publishing Inc.

## FLEX™ USER NOTES

By: Ronald Anderson

The publishers of 68 MICRO JOURNAL are proud to make available the publication of Ron Anderson's **FLEX USER NOTES**, in book form. This popular monthly column has been a regular feature in 68' MICRO JOURNAL SINCE 1979. It has earned the respect of thousands of 68 MICRO JOURNAL readers over the years. In fact, Ron's column has been described as the 'Bible' for 68XX users, by some of the world's leading microprocessor professionals. The most needed and popular 68XX book available. Over the years Ron's column has been one of the most popular in 68 MICRO JOURNAL. And of course 68 MICRO JOURNAL is the most popular 68XX magazine published.

Listed below are a few of the TEXT files included in the book and on diskette.

All TEXT files in the book are on the disks

LOGO.C1	File load program to offset memory — ASM PIC
MEMOVE.C1	Memory move program — ASM PIC
DUMP.C1	Printer dump program — uses LOGO — ASM PIC
SUBTEST.C1	Simulation of 6800 code to 6809, show differences — ASM
TERMEM.C2	Modem input to disk (or other port input to disk) — ASM
M.C2	Output a file to modem (or another port) — ASM
PRINT.C3	Parallel (enhanced) printer driver — ASM
MODEM.C2	TTL output to CRT and modem (or other port) — ASM
SCIPKG.C1	Scientific math routines — PASCAL
U.C4	Mini-monitor, disk resident, many useful functions — ASM
PRINT.C4	Parallel printer driver, without PFLAG — ASM
SET.C5	Set printer modes — ASM
SETBAS1.C5	Set printer modes — A-BASIC

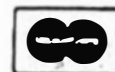
NOTE: .C1, .C2, etc.—Chapter 1, Chapter 2, etc.

\*Over 30 TEXT files included in ASM (assembler)-PASCAL-  
PIC (position independent code) TSC BASIC-C, etc.

Book only: \$7.95 + \$2.50 S/H

With disk: 5" \$20.90 + \$2.50 S/H

With disk: 8" \$22.90 + \$2.50 S/H



(615) 842-4601

Telex 5106006630

# LLOYD I/O

TM

ESTABLISHED 1979

INC.

## ASSEMBLY LANGUAGE DEVELOPMENT SYSTEMS

**VANTAGE**™ is your secret weapon that speeds the design-code-test phases of developing your ideas into real concrete products. **VANTAGE** is a three part package of integrated software including:

**ED**™ ..... Editor  
**CRASMB**™ ..... Macro Cross-Assembler  
**CRACKER**™ ..... Symbolic Debugger

The system speeds up development time because the editor, assembler, and debugger talk to each other directly with out having to exchange data on slow disk files. In fact the only time you need to use the disk is to save your source code when you go home for the day.

**VANTAGE** is compatible with EPROM programmers available on some OS9/68000 computers. EPROM programming is made easy with the built in program, verify, check, copy, and read EPROM commands.

The fast full screen **Editor** is used to code your assembly language program. It is very powerful, including line numbers, source code cross-referencing, 10 edit buffers, command macros, slide file functions, access to OS9, . . . etc. When you are ready to assemble, you simply call the cross assembler. It assembles the source code right out of the editor's text buffer(s).

**CRASMB** is packed with many features to speed your development job. Powerful macro features, symbol cross referencing by line numbers, and plain English error messages are among these. **CRASMB** offers the best in compatibility with the manufacturers' assembly languages. The generated object code is placed directly into the debugger's 64K of target ram, or on a disk file in S1, Intel Hex, or Binary forms. The symbol table remains intact for use by the debugger.

**CRACKER** is chock-full of so many features that it allows you to attack a debugging situation several ways. Since you have just assembled your program, your object code is already in memory.

**CRACKER** features a software simulator, dis-assembler, interactive assembler, memory examine, memory search, dynamic breakpoints and other supportive commands. Breakpoints

are the key in testing software, so we went all out in our research and development of the breakpoint system. We looked at other debuggers on several different computers, and talked with many assembly language programmers. **CRACKER** has easy to use breakpoints, but at the same time they are very powerful, including: conditional breakpoints, unlimited scope, memory protection, many types of actions including the execution of a macro-breakpoint-language for simulation of hardware devices. Expressions may use symbols that were defined in the assembly process to specify memory address values. Breakpoints remember the symbol names making it easy to make changes to your source code without having to kill and enter new breakpoints. The simulator allows you to execute your object code in a controlled environment. If things get out-of-wack, your memory protection and breakpoints will keep the simulation under control. Single stepping displays the registers and either a dis-assembly of the instruction about to be executed or a listing of the source code line from the editor that generated the object code. I.e. source code debugging is built in.

**SERVICE** includes a 1 year free update period from the time of purchase. We ship within 4 working days and use **UPS/BLUE**. Each copy is custom manufactured to your computer disk size and format.

### Supported CPU's

<b>CRASMB</b>	<b>CRACKER</b>
6800,02,08	6800,02,08
6801,6301	6801,6301
6303	6303
6804	6804
6805	6805
6809	6809
6811,H11,HC11	6811,H11,HC11
6502	BY 2Q87
1802	BY 2Q87
TMS7000,20,40	BY 2Q87
8041,8741	BY 3Q87
8048,20,21,22,ETC.	BY 3Q87
8051,31	BY 3Q87
8080,85	BY 3Q87
Z8	BY 3Q87
Z80	BY 4Q87

For OS9/68000 ..... \$795

Includes all Supported CPUs  
(512K Memory Required)

**CRASMB ONLY (OS9/68000)** ..... \$432  
**CRASMB ONLY (OS9/6809)** ..... \$399  
**ED ONLY (OS9/68000)** ..... \$ 63  
**CRACKER and ED ONLY (OS9/68000)** ..... \$495

### DEALER TELEPHONES

**USA**  
SE S.E. Media Supply 1 615 842 4800  
NE Frank Hogg Lab. Inc. 1 315 474 7858  
SW GESPAC, Inc. 1 802 982 5558  
NE Thomas Instrumentation 1 609 967 4280  
NW D & S Industries 1 206 827 7300  
NW Style Software 1 208 529 3210  
**AUSTRALIA**  
Pars Radio Electronics 0 2 344 9111  
**DENMARK**  
H.C. Andersen Computer A/S 0 1 52 44 04  
**ENGLAND**  
Vivaway Ltd. 0 582 423 425  
Windrush Micro Systems 0 692 404 086  
**FRANCE**  
Microdata-Soft 1 768 8080  
Computer Dialysis France 1 4789 8442  
**WEST GERMANY**  
Dr. Rudolf Keil GmbH 0 62 03 87 41  
Zacher Kleincomputer 0 65 25 299  
**JAPAN**  
Seikou Electronics Co. 03 832 8000  
Microboards, Inc. 0 474 22 1741  
**SWEDEN**  
Micromaster Scandinavian AG 018 138585  
**SWITZERLAND**  
Eleoft AG 056 83 33 77



VISA - MASTERCARD



DEALER and OEM inquiries invited  
**VANTAGE, ED, CRASMB, CRACKER**  
are trademarks of LLOYD I/O, INC.

OS9 is a trademark of Microware  
Our Development System  
is a GMX Computer

**LLOYD I/O, INC. • P.O. Box 30945 • Portland, OR 97230 • USA**  
**Attn.: Frank Hoffman • Phone 1-503-666-1097 • Telex 910 380 5448 LLOYD IO**

# SOFTWARE FOR THE **HARDWARE**

.. FORTH PROGRAMMING TOOLS from the 68XX&X ..  
.. FORTH specialists — get the best!! ..

NOW AVAILABLE — A variety of rom and disk FORTH systems to run on and/or do TARGET.COMPIRATION for

6800, 6301/6801, 6809, 68000, 8080, Z80

Write or call for information on a special system to fit your requirement.

Standard systems available for these hardware —

EPSON HX-20 rom system and target compiler  
6809 rom systems for SS-50, EXORCISER, STD, ETC.  
COLOR COMPUTER

6800/6809 FLEX or EXORCISER disk systems.

68000 rom based systems

68000 CP/M-68K disk systems, MODEL II/12/16

tFORTH is a refined version of FORTH Interest Group standard FORTH, faster than FIG-FORTH. FORTH is both a compiler and an interpreter. It executes orders of magnitudes faster than interpretive BASIC. MORE IMPORTANT, CODE DEVELOPMENT AND TESTING is much, much faster than compiled languages such as PASCAL and C. If Software DEVELOPMENT COSTS are an important concern for you, you need FORTH!

firmFORTH\* is for the programmer who needs to squeeze the most into roms. It is a professional programmer's tool for compact rommable code for controller applications.

\* tFORTH and firmFORTH are trademarks of Talbot Microsystems

\* FLEX is a trademark of Technical Systems Consultants, Inc.

\* CP/M-68K is trademark of Digital Research, Inc.

tFORTH™

from TALBOT MICROSYSTEMS

NEW SYSTEMS FOR

6301/6801, 6809, and 68000

—> IFORTH SYSTEMS <—

For all FLEX systems: GIMIX, SWTP, SSB, or EXORCISOR Specify 5 or 8 inch diskette, hardware type, and 6800 or 6809.

\*\* tFORTH — extended fig FORTH (1 disk) \$100 (\$15)  
with fig line editor.

\*\* tFORTH + — more! (3 5" or 2 8" disks) \$250 (\$25)  
adds screen editor, assembler, extended data types, utilities, games, and debugging aids.

\*\* TRS-80 COLORFORTH — available from The Micro Works  
\*\* firm FORTH — 6809 only. \$350 (\$10)

For target compilations to rommable code.

Automatically deletes unused code. Includes HOST system source and target nucleus source. No royalty on targets. Requires but does not include tFORTH +.

\*\* FORTH PROGRAMMING AIDS — elaborate decompiler \$150

\*\* tFORTH for HX-20, in 16K roms for expansion unit or replace BASIC \$170

\*\* tFORTH/68K for CP/M-68K 8" disk system \$290  
Makes Model 16 a super software development system.

\*\* Nautilus Systems Cross Compiler

— Requires: tFORTH + HOST + at least one TARGET:

— HOST system code (6809 or 68000) \$200

— TARGET source code: 6800-\$200, 6301/6801—\$200

same plus HX-20 extensions— \$300

6809—\$300, 8080/Z80—\$200, 68000—\$350

Manuals available separately — price in ( ).  
Add \$6/system for shipping, \$15 for foreign air.

TALBOT MICROSYSTEMS 1927 Curtis Ave., Redondo Beach, CA 90278 (213) 376 9941

## DATA-COMP SPECIAL Heavy Duty Power Supplies

For A limited time we are offering our HEAVY DUTY SWITCHING POWER SUPPLY. These are BRAND NEW units and will not last long. Also note that these prices are less than 1/4 the normal price for these high quality unit.

Make: Boschert

Size: 10.5 x 5 x 2.5 inches - including heavy mounting bracket and terminal.

Rating: in 110/220 volts ac (strap change) Out: 130 watts

Output: +5v - 10 amps  
+12v - 4.0 amps  
+12v - 2.0 amps  
-12v - 0.5 amps

Mating Connector: Terminal strip  
Load Reaction: Automatic short circuit recovery

Each  
SPECIAL: \$59.95  
2 or more 49.95

Add: \$7.50 each S/H

Make: Boschert

Size: 10.75 x 6.2 x 2.25 inches

Rating: 110/220 ac (strap change) Out: 81 watts

Output: +5v - 8.0 amps  
+12v - 2.4 amps  
+12v - 2.4 amps  
+12v - 2.1 amps  
-12v - 0.4 amps

Mating Connector: Molex  
Load Reaction: Automatic short circuit recovery

Each  
SPECIAL: \$49.95  
2 OR MORE 39.95

Add: \$7.50 S/H each



A Division of  
Computer Publishing, Inc.  
5900 Commerce Smith Road  
Hixson, Ga 37343  
Telephone 615 842-4800  
Telex 510 600-6830



## **PL $\mu$ S-68k (PL/9 for the 68000) running under OS/9-68000**

- Built-in screen editor
- Built-in source-level debugger
- Byte, Integer, Long and Real variables
- Direct source to object compilation
- Generates fast, efficient object code
- Compiles over 2000 lines/min
- 99% upward compatible with PL/9
- Comprehensive instruction manual
- No license fees to pay

PL $\mu$ S-68k is a complete re-write of PL/9, for the 68000-series  $\mu$ P. It is a structured high-level language, designed for programming stand-alone applications as well as those in which an operating system is present. It is easy to learn, very quick to use (the compiler is resident with the editor and produces executable object code in one pass) and generates efficient, position-independent code upon which no royalties are payable.

The built-in tracer/debugger helps you debug your programs symbolically and at source level, so you need know next to nothing about the assembly-language of the 68000 in order to develop programs.

PL $\mu$ S is here now, it carries the Windrush commitment to support, and it's cheap. Do you need a better reason to try it?

**PL $\mu$ S-68k for OS/9-68000.....\$299.00**

PL $\mu$ S-68k is also available for FLEX™ systems, supplied with a 68008 second processor, case and power supply. (This reduced price no longer includes a free upgrade to the OS/9 version).....\$499.00

For further information, phone or write:

Worstead Laboratories  
North Walsham  
Norfolk NR28 9SA  
England

Tel (44) 692 404086  
Telex 975548 WMICRO G





# Clearbrook Software Group



## Information Management System

CSG-IMS is a full featured relational database manager with the added benefit of a comprehensive structured application language. This combination makes CGS-IMS the ideal development tool for file intensive applications. Sophisticated applications can be developed in a small fraction of the time required for traditional languages.

- Interactive access to databases and quick ad hoc queries.
- CSG-IMS includes a recursive compiled language supporting program modules with full parameter passing.
- User defined screen and report formats.

### PRICES:

CSG-IMS for OS9 6809 LII	\$495
CGS-IMS for OS9 68000	\$495
CSG-IMS for CoCo3 OS9 (single user)	\$200
CSG-IMS for CoCo (OS9 level 1)	\$150
CSG-IMS demo with manual	\$30
Shipping North America	\$5
Shipping Overseas	\$10



Visa



Mastercard

### ORDER FROM:

Clearbrook Software Group  
P.O. Box 8000-499  
Sumas, WA 98295

(604) 853-9118 BBS (604) 859-1266

OS9 is a trademark of Microware and Motorola Inc.

- Record, index and file size almost unlimited.
- Text, BCD floating point (14 digit), short and long integer and date types.
- Run-time interpreter available.
- Comprehensive 320 page manual/tutorial.

## !!! Subscribe Now !!! 68 MICRO JOURNAL

### Subscription Rates

U.S.A.: 1 Year \$24.50, 2 Years \$42.50, 3 Years \$64.50

\*Foreign Surface: Add \$12.00 per Year to USA Price.

\*Foreign Airmail: Add \$48.00 per Year to USA Price.

\*Canada & Mexico: Add \$9.50 per Year to USA Price.

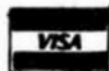
\*U.S. Currency Cash or Check Drawn on a USA Bank !

68 Micro Journal  
5900 Cassandra Smith Rd.  
POB 849  
Hixson, TN 37343



Telephone 615 842-4600

Telex 510 600-6630



### OK, PLEASE ENTER MY SUBSCRIPTION

Bill My: Mastercard ☐ VISA ☐

Card # \_\_\_\_\_ Exp. Date \_\_\_\_\_

For 1 Year \_\_\_\_\_ 2 Years \_\_\_\_\_ 3 Years \_\_\_\_\_

Enclosed: \$ \_\_\_\_\_

Name \_\_\_\_\_

Street \_\_\_\_\_

City \_\_\_\_\_ State \_\_\_\_\_ Zip \_\_\_\_\_

My Computer Is: \_\_\_\_\_

## OS-9™ SOFTWARE

**L1 UTILITY PAK**—Contains all programs formerly in Filter kits 1 & 2, and Hacker's kit 1 plus several additional programs. Complete "wild card" file operations, copies, moves, sorts, del, MACGEN shell command language compiler, Disassembler, Disk sectoredit utility, new and improved editions, approx. 40 programs, increases your productivity. Most programs applicable for both level I & II 6809 OS-9. **\$49.95 (\$51.95)**

Call or send Self Addressed Stamped Envelope for catalog of software for color Computer OS-9 and other OS-9 systems.

BOLD prices are CoCo OS-9 format disk, other formats (in parenthesis) specify format. All orders prepaid or COD, VISA and MasterCard accepted. Add \$1.50 S&H on prepaid, COD actual charges added.

### SS-50C MEMORY LIQUIDATION SALE! (While Supply Lasts)

#### 1 MEGABYTE RAM BOARD

Full megabyte of ram with disable options to suit any SS-506809 system. High reliability, can replace static ram for fraction of the cost. \$399 for 2 Mhz or \$439 for 2.25 Mhz board assembled, tested and fully populated.

#### 2 MEGABYTE RAM DISK BOARD

RD22 megabytes dedicated ram disk board for SS-50 systems. Four layer circuit board socketed for 2 **Megabytes!** Special sale price of **\$399.00** includes only 256k of ram installed (you add the rest), includes OS-9 level I and II drivers for Ram disk, (note: you can reboot your system without losing ram-disk contents). (Add \$6 shipping and insurance.)

Please call for answers to your technical questions concerning these products.

D.P. Johnson, 7655 S.W. Cedarcrest St.  
Portland, OR 97223, (503) 244-8152  
(For best service call between 9-11 am Pacific time.)

OS-9 is a trademark of Microwave and Motorola Inc.  
MS-DOS is a trademark of Microsoft Inc.

## COMPILER EVALUATION SERVICES

BY: Ron Anderson

The S.E. MEDIA Division of Computer  
Publishing Inc.  
is offering the following **SUBSCRIBER  
SERVICE:**

### COMPILER COMPARISON AND EVALUATION REPORT

Due to the constant and rapid updating and enhancement of numerous compilers, and the different utility, appeal, speed, level of communication, memory usage, etc., of different compilers, the following services are now being offered with periodic updates.

This service, with updates, will allow you who are wary or confused by the various claims of compiler vendors, an opportunity to review comparisons, comments, benchmarks, etc., concerning the many different compilers on the market, for the 6809 microcomputer. Thus the savings could far offset the small cost of this service.

Many have purchased compilers and then discovered that the particular compiler purchased either is not the most efficient for their purposes or does not contain features necessary for their application. Thus the added expense of purchasing additional compiler(s) or not being able to fully utilize the advantages of high level language compilers becomes too expensive.

The following **COMPILERS** are reviewed initially, more will be reviewed, compared and benchmarked as they become available to the author:

PASCAL 'C' GSPL WHIMSICAL PL/9

Initial Subscription - \$39.95

(includes 1 year updates)

Updates for 1 year - \$14.50

S.E. MEDIA - C.P.I.  
5900 Cassandra Smith Rd.  
Hixson, TN 37343  
(615) 842-4601

# E 68000 68020 68010

# 68008 68009 68000

Write or phone for catalog.

## AAA Chicago Computer Center

120 Chestnut Lane — Wheeling IL 60090  
(312) 459-0450

Technical Consultation available most weekdays from 4 PM to 8 PM CST

# Stop!

**Get a 25  
MegaByte Hard  
Disk practically  
FREE - only 1¢**

**Be Sure to Consider the  
SPECIAL MUSTANG  
1¢ Sale on page 5**

**When it's over, IT'S OVER!**

We don't know how long this very, very low price can be maintained, don't miss it!

Data-Comp Div. - CPI

# 6809<>68XXX UniFLEX

## X-TALK

### A C-MODEM/Hardware Hookup

Exclusive for the MUSTANG-020 running UniFLEX, is a new transfer program and cable set from DATA-COMP (CPI). X-TALK consist of 2 disks and a special cable, this hook-up enables a 6809 SWTPC UniFLEX computer to port UniFLEX files directly to a 68XXX UniFLEX system.

This is the only currently available method to transfer files, text or otherwise, from a 6809 UniFLEX system to a 68000 UniFLEX system, that we have seen. A must if you want to recompile or cross assemble your old (and valuable) source files to run on a 68000 UniFLEX system. GIMIX users can directly transfer files between a 6809 GIMIX system and our MUSTANG-020 68020 system, or GIMIX 68020 system. All SWTPC users must use some sort of method other than direct disk transfer. The 6809 SWTPC UniFLEX disk format is not readable by most other 68000 type systems.

The cable is specially prepared with internal connections to match the non-standard SWTPC SOV9 DB25 connectors. A special SWTPC+ cable and software is also available, at the same price. Orders must specify which type SWTPC 6809 UniFLEX system they intend to transfer from or to.

The X-TALK software is furnished on two disks. One 8" disk containing the 6809 software and one 5" disk containing the 68XXX software. These programs are also complete MODEM programs and can be used as such, including X-on X-off, and all the other features you would expect from a full modem program.

X-TALK can be purchased with/without the special cables, however, this SPECIAL price is available only to registered MUSTANG-020 owners.

**X-TALK, w/cable \$ 99.95**  
**X-TALK only 69.95**  
**X-TALK w/source \$149.95**

**DATA-COMP**  
5900 Cassandra Smith Rd.  
Hixson, TN 37343

Telephone 615 842-4601  
Telex 510 600-6630

Note: Registered MUSTANG-020 owners must furnish system serial number in order to buy at these special low prices.

## 68 MICRO JOURNAL Reader Service Disks

- Disk- 1 Filesort, Minicat, Minicopy, Minifms, \*\*Lifetime, \*\*Poetry, \*\*Foodlist, \*\*Diet.
- Disk- 2 Diskedit w/ inst. & fixes, Prime, \*Prmod, \*\*Snoopy, \*\*Football, \*\*Hexpaw, \*\*Lifetime.
- Disk- 3 Cbug09, Sec1, Sec2, Find, Table2, Intext, Disk-exp, \*Disksave.
- Disk- 4 Mailing Program, \*Finddat, \*Change, \*Testdisk.
- Disk- 5 \*DISKFIX 1, \*DISKFIX 2, \*\*LETTER, \*\*LOVESIGN, \*\*BLACKJAK, \*\*BOWLING.
- Disk- 6 \*\*Purchase Order, Index (Disk file indx).
- Disk- 7 Linking Loader, Rload, Harkness.
- Disk- 8 Crest, Lanpher (May 82).
- Disk- 9 Datacopy, Diskfix9 (Aug 82).
- Disk-10 Home Accounting (July 82).
- Disk-11 Disassembler (June 84).
- Disk-12 Modem68 (May 84).
- Disk-13 \*Initmf68, Testmf68, \*Cleanup, \*Dakalign, Help, Data.Txt.
- Disk-14 \*Init, \*Test, \*Terminal, \*Find, \*Diskedit, Init.Lib
- Disk-15 Modem9 + Updates (Dec. 84 Gilchrist) to Modem9 (April 84 Commo).
- Disk-16 Copy.Txt, Copy.Doc, Cat.Txt, Cat.Doc.
- Disk-17 Match Utility, RATBAS, A Basic Preprocessor.
- Disk-18 Parse.Mod, Size.Cmd (Sept. 85 Armstrong), CMDC ODB, CMD.Txt (Sept. 85 Spray).
- Disk-19 Clock, Date, Copy, Cat, PDEL.Asm & Doc, Errors.Sys, Do, Log.Asm & Doc.
- Disk-20 UNIX Like Tools (July & Sept. 85 Taylor & Gilchrist), Dragon.C, Grep.C, LSC, FDUMP.C.
- Disk-21 Utilities & Games - Date, Life, Madness, Touch, Goblin, Starshot, & 15 more.
- Disk-22 Read CPM & Non-FLEX Disks. Fraser May 1984.
- Disk-23 ISAM, Indexed Sequential file Accessing Methods, Condon Nov. 1985. Extensible Table Driven. Language Recognition Utility, Anderson March 1986.
- Disk-24 68' Micro Journal Index of Articles & Bit Bucket Items from 1979 - 1985, John Current.
- Disk-25 KERMIT for FLEX derived from the UNIX ver. Burg Feb. 1986. (2)-5" Disks or (1)-8" Disk.
- Disk-26 Compact UniBoard review, code & diagram, Buisson March '86.
- Disk-27 ROTABIT.TXT, SUMTEST.TXT, CONDATA.TXT, BADMEN.TXT.
- Disk-28 CT-82 Emulator, bit mapped.
- Disk-29 \*\*Star Trek
- Disk-30 Simple Winchester, Dec.'86 Green.
- Disk-31 \*\*\* Read/Write MS/PC-DOS (SK'DOS)
- Disk-32 Ilier-UNIX Type upgrade - 68MJ 2/87

### NOTE:

This is a reader service ONLY! No Warranty is offered or implied, they are as received by 68' Micro Journal, and are for reader convenience ONLY (some MAY include fixes or patches). Also 6800 and 6809 programs are mixed, as each is fairly simple (mostly) to convert to the other. Software is available to cross-assemble all.

\* Denotes 6800 - \*\* Denotes BASIC  
\*\*\* Denotes 68000 - 6809 no indicator.



Specify 8" disk \$19.50  
5" disk \$16.95



Add: S/H - \$3.50

Overseas add: \$4.50 surface - \$7.00 Air Mail, USA Dollars

## 68 MICRO JOURNAL

PO Box 849

Hixson, TN 37343

615 842-4600 - Telex 510 600-6630

## 6809/68008 SINGLE BOARD COMPUTERS

The Peripheral Technology Family of Single Board Computers is a Low-Cost Group Which Ranges From an Entry Level 8-Bit Version to a Powerful 68008-Based Board. A Product is Available to Fit Almost Every User's Requirements.

### PT69-5

- 6809 Processor/2MHz Clock
- 4 RS-232 Serial Ports
- 2 8-Bit Parallel Ports
- 4K-16K EPROM/60K Ram
- Parallel Printer Interface
- DS/DD Controller for 35-80
- Track Drives Ranging From SS/SD-DS/DD
- Winchester Interface Port

PRICE: \$349.00



### PT69-3

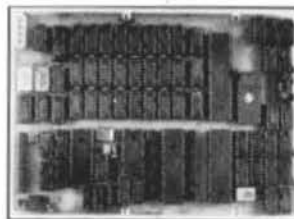
- 6809 1 MHz Processor
- 2 RS-232 Serial Ports
- 2 8-Bit Parallel Ports
- 4K EPROM/59K User Ram
- DS/DD Controller for 35-80 Track Drives Ranging From SS/SD-DS/DD

PRICE: \$269.95  
OS9 L1 For  
PT69 BOARDS: \$200.00  
SK'DOS: \$49.95

### PT68K-1

- MC68008 10 MHz Processor
- 768K RAM/64K EPROM
- 2 RS-232 Serial Ports
- Winchester Interface Port
- Floppy Disk Controller for 2 5 1/4" Drives
- 2 8-Bit Parallel Ports

BOARD: \$595.00  
WITH OS9: \$749.95  
WITH SK'DOS: \$675.00



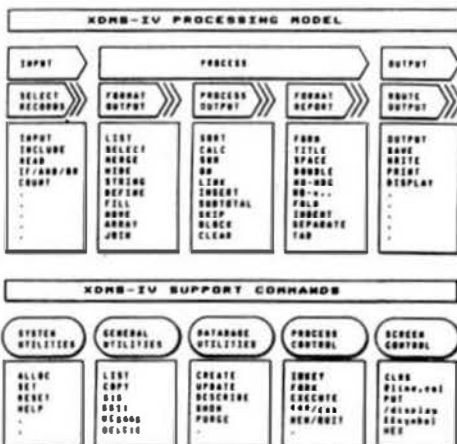
**PERIPHERAL TECHNOLOGY**  
1480 Terrell Mill Road, Suite 870  
Marietta, Georgia 30067  
(404) 984-0742 Telex # 880584  
VISA/MASTERCARD/CHECK/C.O.D.

\*OS9 is A Trademark Of Microware and Motorola.

Send For Catalogue For Complete Information On All Products.

## XDMS-IV

### Data Management System



Save \$100.00 - Limited Time  
Regular \$350.00 - Now Only  
**\$249.95**

### FOR 6809 FLEX-SK-DOS(5/8")

Up to 32 groups/fields per record! Up to 12 character field name! Up to 1024 byte records! User defined screen and print control! Process files! Form files! Conditional execution! Process chaining! Upward/Downward file linking! File joining! Random file virtual paging! Built in utilities! Built in text editor! Fully session oriented! Enhanced forms! Boldface, Double width, Italics and Underline supported! Written in compact structured assembler! Integrated for FAST execution!

#### XDMS-IV Data Management System

XDMS-IV is a brand new approach to data management. It not only permits users to describe, enter and retrieve data, but also to process entire files producing customized reports, screen displays and file output. Processing can consist of any of a set of standard high level functions including record and field selection, sorting and aggregation, lookups in other files, special processing of record subsets, custom report formatting, totaling and subtotaling, and presentation of up to three related files as a "database" on user defined output reports.

#### POWERFUL COMMANDS!

XDMS-IV combines the functionality of many popular DBMS software systems with a new easy to use command set into a single integrated package. We've included many new features and commands including a set of general file utilities. The processing commands are Input-Process-Output (IPO) oriented which allows almost instant implementation of a process design.

#### SESSION ORIENTED!

XDMS-IV is session oriented. Enter "XDMS" and you are in instant command of all the features. No more waiting for a command to load in from disk! Many commands are immediate, such as CREATE (file definition), UPDATE (file editor), PURGE and DELETE (utilities). Others are process commands which are used to create a user process which is executed with a RUN command. Either may be entered into a "process" file which is executed by an EXECUTE statement. Processes may execute other processes, or themselves, either conditionally or unconditionally. Menus and screen prompts are easily coded, and entire user applications can be run without ever leaving XDMS-IV!

#### IT'S EASY TO USE!

XDMS-IV keeps data management simple! Rather than design a complex DBMS which hides the true nature of the data, we kept XDMS-IV file oriented. The user view of data relationships is presented in reports and screen output, while the actual data resides in easy to maintain files. This aspect permits customized presentation and reports without complex redefinition of the database files and structure. XDMS-IV may be used for a wide range of applications from simple record management systems (addresses, inventory ...) to integrated database systems (order entry, accounting...). The possibilities are unlimited...

Visa & Master Card Excepted

Telephone: 615-842-4601 or Telex: 510 600-6630  
Or Write: S.E. Media, 5900 Cassandra Smith Rd.,  
Hixson, Tenn. 37343



Technical telephone assistance: Tel 914-941-3552 (Evenings)  
FLEX™ Technical Systems Consultants, SK'DOS™ STAR-KITS Corp.



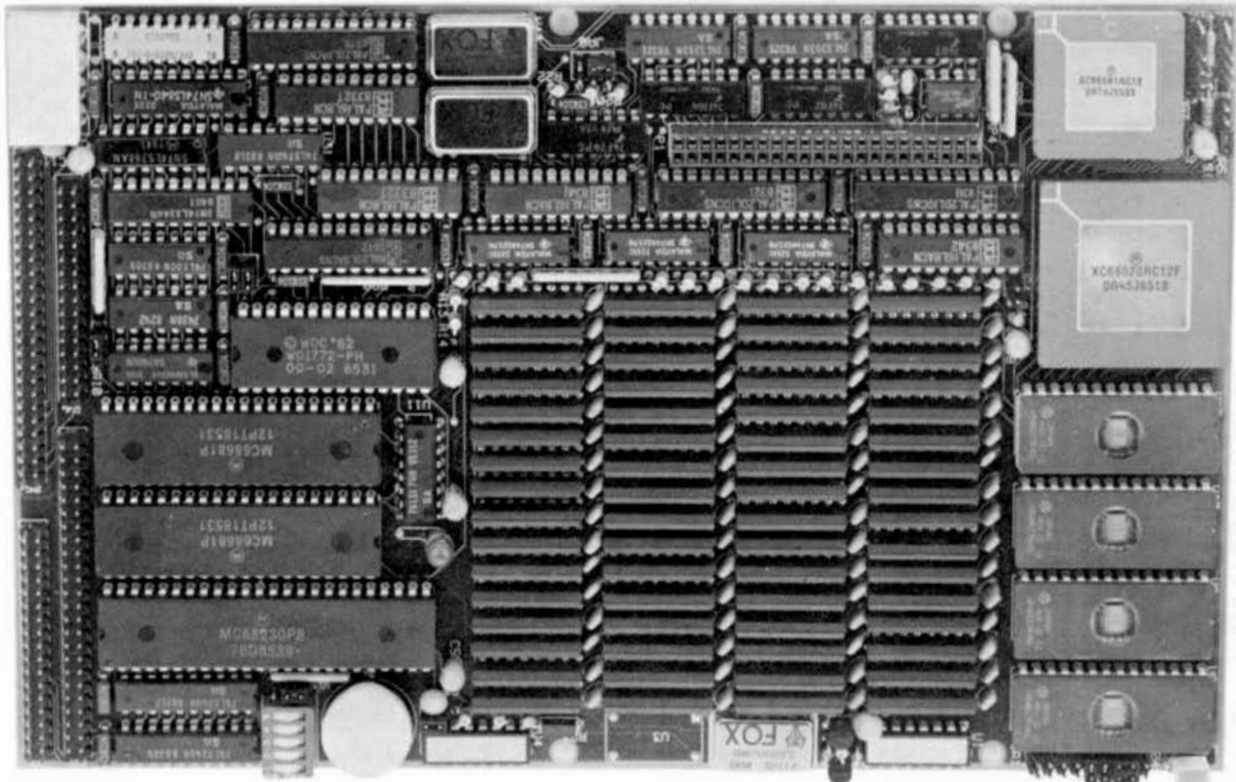
# GMX™ Micro-20

## 68020 SINGLE-BOARD COMPUTER

**Mainframe CPU Performance**  
**on a 5.75" x 8.8" Board**  
 (benchmark results available on request)

**\$2565<sup>00</sup>**

12.5 MHz Version  
 Quantity Discounts Available



### Features

- 32-Bit MC68020 Processor (12.5, 16.67, or 20MHz)
- MC68881 Floating-point coprocessor (optional)
- 2 Megabytes of 32-bit wide, high-speed RAM
- 4 RS-232 Serial I/O Ports (expandable to 36)
- 8-bit Parallel I/O Port (Centronics' compatible)
- Time-of-Day Clock w/ battery backup
- 16-bit I/O Expansion Bus
- Up to 256 Kbytes of 32-bit wide EPROM
- Floppy Disk Controller for two 5 1/4" drives
- SASI Intelligent Peripheral Interface (SCSI subset)
- Mounts directly on a 5 1/4" Disk Drive
- Optional Boards include Arcnet, Prototyping, I/O Bus adapter, 60 line Parallel I/O, RS-422/485

### Software

#### Included:

- GMX Version of Motorola's 020Bug Debugger with up/download, breakpoint, trace, single-step, and assembler/disassembler capabilities
- Comprehensive Hardware Diagnostics

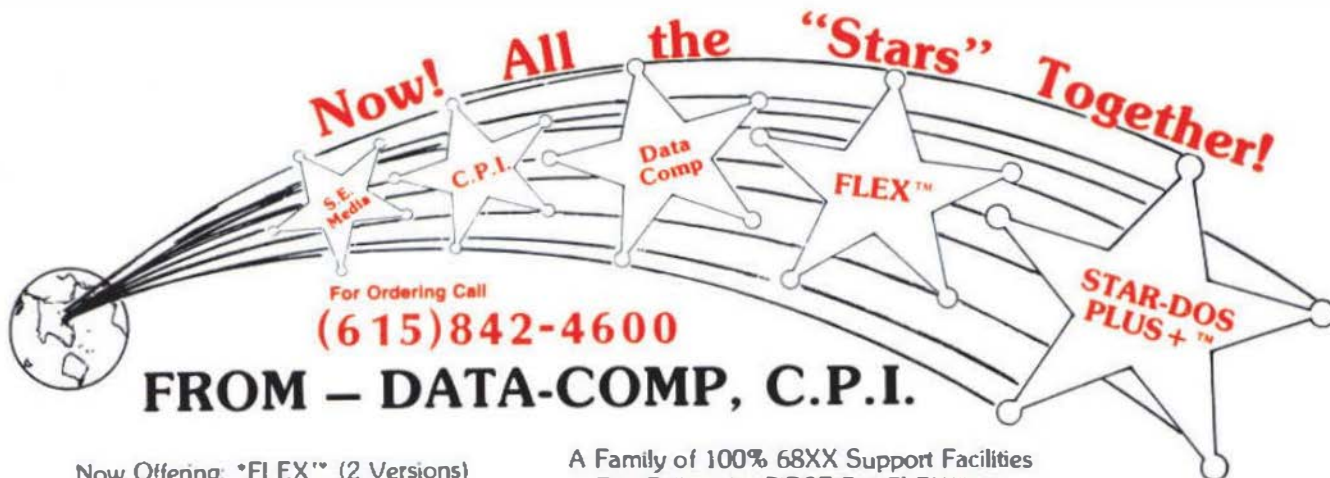
#### Optional:

- *UNIX™ like Multi-user/Multi-tasking Disk Operating Systems*
  - OS-9/68000™ (Real-time and PROMable)
  - UniFLEX™  
*Programming Languages and Application Software*
  - BASIC, C, PASCAL, ABSOFT FORTRAN, COBOL and ASSEMBLER
  - Spreadsheet, Data Base Management, and Word Processing
- COMPLETE EVALUATION SYSTEMS AVAILABLE**

**GMX™** 1337 W. 37th Place Chicago, IL 60609

(312) 927-5510 • TWX 910-221-4055  
*State-of-the-Art Computers*  
 Since 1975





Now Offering: \*FLEX\* (2 Versions)  
AND \*STAR-DOS PLUS+™

A Family of 100% 68XX Support Facilities  
The Folks who FIRST Put FLEX™ on  
The CoCo

**FLEX-CoCo Sr.**  
with TSC Editor  
TSC Assembler

Complete with Manuals  
Reg. \$250.<sup>00</sup> **Only \$79.<sup>00</sup>**

**STAR-DOS PLUS+**

- Functions Same as FLEX
- Reads - writes FLEX Disks **\$34.<sup>00</sup>**
- Run FLEX Programs
- Just type: Run "STAR-DOS"
- Over 300 utilities & programs to choose from.

**FLEX-CoCo Jr.**  
without TSC  
Editor & Assembler  
**\$49.<sup>00</sup>**

**PLUS**

**ALL VERSIONS OF FLEX & STAR-DOS INCLUDE**

**TSC Editor**

Reg \$50.00

**NOW \$35.00**

- + Read-Write-Dir RS Disk
- + Run RS Basic from Both
- + More Free Utilities

- + External Terminal Program
- + Test Disk Program
- + Disk Examine & Repair Program
- + Memory Examine Program
- + Many Many More!!!

**TSC Assembler**

Reg \$50.00

**NOW \$35.00**

**CoCo Disk Drive Systems**

2 THINLINE DOUBLE SIDED DOUBLE DENSITY DISK DRIVES  
SYSTEM WITH POWER SUPPLY, CABINET, DISK DRIVE CABLE, J&M  
NEW DISK CONTROLLER JPD-CP WITH J-DOS, RS-DOS OPERATING  
SYSTEMS. **\$469.95**

\* Specify What CONTROLLER You Want J&M, or **RADIO SHACK**

THINLINE DOUBLE SIDED  
DOUBLE DENSITY 40 TRACKS **\$129.95**

**Verbatim Diskettes**

Single Sided Double Density **\$ 24.00**  
Double Sided Double Density **\$ 24.00**

**Controllers**

J&M JPD-CP WITH J-DOS **\$139.95**  
WITH J-DOS, RS-DOS **\$159.95**  
RADIO SHACK 1.1 **\$134.95**

RADIO SHACK Disk CONTROLLER 1.1 **\$134.95**

**Disk Drive Cables**

Cable for One Drive **\$ 19.95**  
Cable for Two Drives **\$ 24.95**

**MISC**

64K UPGRADE **\$ 29.95**  
FOR C,D,E,F, AND COCO 11  
RADIO SHACK BASIC 1.2 **\$ 24.95**  
RADIO SHACK DISK BASIC 1.1 **\$ 24.95**

DISK DRIVE CABINET FOR A  
SINGLE DRIVE **\$ 49.95**  
DISK DRIVE CABINET FOR TWO  
THINLINE DRIVES **\$ 69.95**

**PRINTERS**

EPSON LX-80 **\$289.95**  
EPSON MX-70 **\$125.95**  
EPSON MX-100 **\$495.95**

**ACCESSORIES FOR EPSON**

8148 2K SERIAL BOARD **\$ 89.95**  
8149 32K EXPAND TO 128K **\$169.95**  
EPSON MX-RX-80 RIBBONS **\$ 7.95**  
EPSON LX-80 RIBBONS **\$ 5.95**  
TRACTOR UNITS FOR LX-80 **\$ 39.95**  
CABLES & OTHER INTERFACES  
CALL FOR PRICING

**DATA-COMP**

5900 Cassandra Smith Rd.  
Hixson, TN 37343



**SHIPPING**  
USA ADD 2%  
FOREIGN ADD 5%  
MIN. \$2.50

**(615)842-4600**

For Ordering  
Telex 5108008630

# Introducing

## S - 50 BUS / 68XX

Board and/or Computer  
Terminals-CRTs-Printers  
Disk Drives-etc.

## REPAIRS



### NOW AVAILABLE TO ALL S50/68XX USERS

The Data-Comp Division of CPI is proud to announce the availability of their service department facilities to 'ALL' S50 Bus and 68XX users. Including all brands, SWTPC - GIMIX - SSB - HELIX and others, including the single board computers. \*Please note that kit-built components are a special case, and will be handled on an individual basis, if accepted.

↑  
**This**

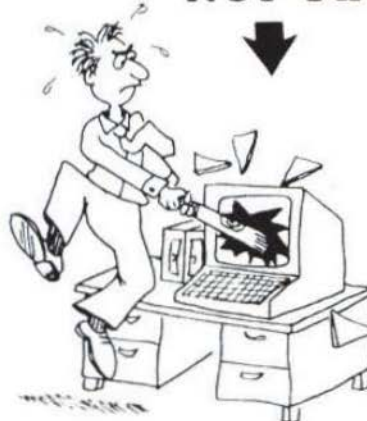
1. If you require service, the first thing you need to do is call the number below and describe your problem and *confirm a Data-Comp service & shipping number!* This is very important, Data-Comp will not accept or repair items not displaying this number! Also we cannot advise or help you troubleshoot on the telephone, we can give you a shipping number, but **NO** advice! Sorry!

2. All service shipments must include both a minimum \$40.00 estimate/repair charge and pre-paid return shipping charges (should be same amount you pay to ship to Data-Comp).

3. If you desire a telephone estimate after your repair item is received, include an additional \$5.00 to cover long distance charges. Otherwise an estimate will be mailed to you, if you requested an estimate. Estimates *must be requested*. Mailed estimates slow down the process considerably. However, if repairs are not desired, after the estimate is given, the \$40.00 shall constitute the estimate charge, and the item(s) will be returned unrepaid providing sufficient return shipping charges were included with the item to be serviced. Please note that estimates are given in dollar amounts only.

4. Data-Comp service is the oldest and most experienced general S50/68XX service department in the world. We have over \$100,000.00 in parts in stock. We have the most complete set of service documents for the various S50/68XX systems of anyone - **YET, WE DO NOT HAVE EVERYTHING!** But we sure have more than anyone else. We repair about 90% of all items we receive. Call for additional information or shipping instructions.

**Not This**



**DATA-COMP**  
5900 Cassandra Smith Rd.  
Hixson, TN 37343

(615)842-4607  
Telex 5106006630

